

**SHELLS AND SPHERES: A NOVEL FRAMEWORK
FOR VARIABLE SCALE STATISTICAL IMAGE
ANALYSIS**

by

C. Aaron Cois

Bachelor of Science, University of Pittsburgh, 2003

Submitted to the Graduate Faculty of
the Department of Computer Science in partial fulfillment
of the requirements for the degree of
Master of Sciences

University of Pittsburgh

2006

UNIVERSITY OF PITTSBURGH
COMPUTER SCIENCE DEPARTMENT

This thesis was presented

by

C. Aaron Cois

It was defended on

May 9th 2006

and approved by

George Stetten, Bioengineering

Milos Hauskrecht, Computer Science

C. C. Li, Computer Science

Thesis Advisors: George Stetten, Bioengineering,

Milos Hauskrecht, Computer Science

Copyright © by C. Aaron Cois

2006

SHELLS AND SPHERES: A NOVEL FRAMEWORK FOR VARIABLE SCALE STATISTICAL IMAGE ANALYSIS

C. Aaron Cois, M.S.

University of Pittsburgh, 2006

A framework for analyzing images, called *Shells and Spheres*, has been developed based on a set of spheres with adjustable radii, with exactly one sphere centered at each image pixel. This set of spheres, known as a *sphere map*, is considered optimized when each sphere reaches, but does not cross, the nearest boundary. Calculations denoted as *Variable-Scale Statistics* (VSS) are performed on populations of pixels within spheres, as well as populations of adjacent and overlapping spheres, in order to deduce the proper radius of each sphere. Spheres grow or shrink by adding or deleting an outer shell one pixel thick. Unlike conventional fixed-scale kernels, our spherical operators consider as many pixels as possible to differentiate between objects and accurately delineate boundaries. The term “sphere” is used for brevity, though the approach is not limited to 3D and is valid in n -dimensions. The approach is illustrated using both real images and noiseless synthetic images containing objects with uniform intensity, and more closely examined and validated using various synthetic images with added white noise and multiple contrast enhanced CT scans of the aortic arch. A particular algorithm using Shells and Spheres is described and demonstrated on segmentation of the aortic arch in a contrast-enhanced CT scan, both in 2D and 3D.

TABLE OF CONTENTS

1.0 INTRODUCTION	1
2.0 BACKGROUND	3
2.1 Scale Space	3
2.2 Distance Maps	4
2.3 Fuzzy-Connectedness	5
2.4 Mathematical Morphology	7
2.5 Medialness	8
3.0 METHODOLOGY	10
3.1 Shells and Spheres Notation	10
3.2 Variable Scale Statistics	13
3.2.1 Primary Statistics	13
3.2.2 Secondary Statistics	14
3.2.3 Demonstration on a Noiseless Image	17
4.0 ALGORITHM FOR SEGMENTING REAL IMAGES	19
4.1 Step 1: VSS Gradient-Based Radius Approximation	21
4.2 Step 2: Variance-Constrained Radius Reduction	24
4.3 Step 3: Outpost Selection and Exclusion	27
4.4 Step 4: Variance-Constrained Scale Growth	33
4.5 Step 5: Medial Pixel Identification	35
4.6 Step 6: Medial Flood-Fill Segmentation	36
5.0 RESULTS	38
6.0 CONCLUSIONS	46

7.0 ACKNOWLEDGEMENTS	48
BIBLIOGRAPHY	49

LIST OF FIGURES

1	The Blum medial manifold (dotted line) of a rectangle, with selected medial spheres shown.	8
2	Each pixel is shown as a number indicating its integer distance from the central pixel. If we denote the central pixel as \mathbf{x} , then pixels labeled with integer n are members of the set $H_n(\mathbf{x})$. For example, the pixels labeled “3” (shown in bold) comprise the set $H_3(\mathbf{x})$	12
3	Noiseless image with boundary between two objects. Correctly scaled spheres $S_r(\mathbf{x})$ with $r(\mathbf{x}) = 3$ and $S_r(\mathbf{y})$ with $r(\mathbf{y}) = 2$ touch, but do not cross, the boundary. Numbers indicate pixel intensity.	12
4	Correctly optimized sphere map of the image in Fig. 3. Numbers indicate the integer radius of the sphere at each pixel. Pixels \mathbf{x} and \mathbf{y} are labeled as before.	13
5	Set $S^{-1}(\mathbf{x})$ of spheres that contain pixel \mathbf{x} , adjacent to the boundary between two noiseless objects with respective intensities of 1 and 9. Numbers indicate the intensity value of the pixel. (centers of the spheres shown in bold)	15
6	Image with noise. Numbers indicate the intensity value of the pixel. Pixel \mathbf{x} is deterred from extending its sphere across the boundary because its mean is an outlier in the population $S^{-1}(\mathbf{y})$	16
7	A: A noiseless synthetic image. B: The optimal radius image. C: The VSS gradient in the y direction, calculated using spheres of radius 5 pixels at all image points. D: The variance image, also calculated with all spheres set to a radius of 5.	18

8	Images used for illustrative testing. Left: A 2D synthetic image with homogeneous regions, including added synthetic uniform noise (calculated using Equation 5.1, detailed in the Results chapter). Right: A contrast-enhanced CT scan of the aortic arch.	19
9	A: The sphere map $r(\mathbf{x})$ of the image after Step 1. B: The variance of each pixel after Step 1. The grayscale range of the image is 0-300, while all green pixels are distinctly higher than this range.	22
10	A colored height map of the variance image from Fig. 9B	23
11	A: Sphere map $r(\mathbf{x})$ after Step 1. B: The variance of each pixel after Step 1. The grayscale range of the image is 0-915, while all green pixels are distinctly higher than this range.	23
12	A: The sphere map $r(\mathbf{x})$ of the image after Step 2. B: Variance image after Step 2. The grayscale range of the image is 0-300.	25
13	A colored height map of the variance image from Fig. 12B, showing the state of variance after Step 2.	26
14	A: Sphere map $r(\mathbf{x})$ after Step 2. B: An image of the variance of each pixel after Step 2. The grayscale range of the image is 0-915.	26
15	Illustration of $K^{-1}(\mathbf{x})$ containing 7 pixels (bold), each of whose sphere would place its reflector across the boundary at \mathbf{x}	28
16	A: Reflector count of all pixels after Step 3 with a binary threshold ($ K^{-1}(\mathbf{x}) \geq 1$). B: Reflector count of all pixels after Step 3 with $ K^{-1}(\mathbf{x}) \geq 4$. C: The radius image $r(\mathbf{x})$ after Step 3. D: Variance after Step 3, with a grayscale range of 0-300.	30
17	A: Reflector count of all pixels after Step 3 with a binary threshold ($ K^{-1}(\mathbf{x}) \geq 1$). B: Reflector count of all pixels after Step 3 with $ K^{-1}(\mathbf{x}) \geq 4$. C: The radius image $r(\mathbf{x})$ after Step 3. D: Variance after Step 3, with a grayscale range of 0-915.	31
18	Left: Primary (red) and Secondary (yellow) outposts calculated for the synthetic test image. Right: Primary (red) and Secondary (yellow) outposts calculated for the real test image.	32

19	A: Reflector count of all pixels after Step 4 with a binary threshold ($ K^{-1}(\mathbf{x}) \geq 1$) B: Reflector count of all pixels after Step 4 with a binary threshold $ K^{-1}(\mathbf{x}) \geq 4$. C: Radius image $r(\mathbf{x})$ of the image after Step 4. D: Variance after Step 4, with a grayscale range of 0-300.	33
20	A: Reflector count of all pixels with a binary threshold ($ K^{-1}(\mathbf{x}) \geq 1$) B: Reflector count of all pixels after Step 4 with a binary threshold $ K^{-1}(\mathbf{x}) \geq 4$. C: Radius image $r(\mathbf{x})$ of the image after Step 4. D: Variance after Step 4, with a grayscale range of 0-915.	34
21	Diagram on the left of an object with intensity 1 between two regions of intensity 9, showing set $S^{-1}(\mathbf{b})$ of pixels (bold) whose spheres contain pixel \mathbf{b} . This set produces an $\mathbf{s}(\mathbf{b})$ vector along which the furthest bold pixel \mathbf{m} is the center of a medial sphere (circle in bold) touching both boundaries (dashed lines). Image on the right of a 2D slice through a CT scan of the aorta with contrast shows an actual $S^{-1}(\mathbf{b})$ set (purple/grey), the resulting $\mathbf{s}(\mathbf{b})$ vector, and the medial manifold (dashed curve) on which the furthest sphere along $\mathbf{s}(\mathbf{b})$ must lie.	35
22	A: 2D CT image of the aortic arch. B: Optimized $r(\mathbf{x})$ image. C: Measured mean of means $\mu_{\mu}(\mathbf{x})$ image given optimized $r(\mathbf{x})$. D: Segmentation of aortic arch (pink) and medial pixels within the image objects (blue).	39
23	Surface model of the aortic arch produced by applying our algorithm to a 3D contrast-enhanced CT scan, using a single manually placed seed point.	40
24	Left: A different 3D contrast-enhanced CT scan of the aortic arch. Right: An overlaid surface model of the aortic arch produced by applying our algorithm, using a single manually placed seed point.	40
25	Left: A binary synthetic branching structure image. Middle: The radius image $r(\mathbf{x})$ resulting from optimization using the algorithm presented. Right: A subsequent segmentation of the image, showcasing the medial pixels detected (highlighted in yellow). (DSC = 0.999927)	41
26	Synthetic binary image used to test segmentation performance in the presence of noise.	42

27	Synthetic test image with increasing levels of additive noise, split into two columns of test data. The leftmost images in each column are the noisy images, with the amplitude of the additive noise marked in the lower left-hand corner. The middle images in each column are their respective computed radius images. The rightmost images in each column are the segmentations produced along with highlighted medial pixels (yellow), labeled with DSC similarity values for the segmentation.	43
28	Graph of the amplitude of noise added to the test images versus the segmentation performance, measured by DSC similarity to the known segmentation.	44

1.0 INTRODUCTION

The framework of Shells and Spheres described in this thesis is based on a construct called a *sphere map*. A sphere map is a set of spheres, exactly one sphere centered at each image pixel, whose radii can be adjusted. Calculations denoted as *Variable-Scale Statistics* (VSS) are performed on populations of pixels within spheres, as well as populations of adjacent and overlapping spheres. Memory and computational requirements are kept reasonable by storing only a relatively small, fixed number of VSS at each pixel, many of which can be updated incrementally as spheres grow or shrink. The ultimate goal of radii adjustment is to produce a sphere map in which each sphere is as large as possible without crossing a boundary. Once properly adjusted, therefore, the spheres' radii are equivalent to what is commonly known as a *distance map* [1]. Though the task is trivial in binary images, where definitive boundaries are known, it presents a challenge when boundaries are difficult to determine due to noise and tissue inhomogeneity. The Shells and Spheres approach is well suited to this challenge, as it makes use of uniquely large pixel populations defined by maximally sized spheres for its region-defining statistics, though it is understood that the correctness of segmentation of real images is ultimately subjective.

Many conventional methods for image processing consider a region of fixed size and shape, usually referred to as a *kernel*, especially when used for convolution. Other common approaches define dynamic regions adjoining boundaries using deformable contours [2] or level sets [3]. Our approach, instead, uses a set of spheres whose individual radii are optimized using VSS operators to achieve maximum discrimination between image regions. Not only do such spheres provide highly representative populations for boundary detection and region representation, but those spheres that touch at least two boundaries are also *medial*, as classically defined by Blum [4], providing a basis for medial feature extraction. Unlike

Gaussian blurring, commonly used in multi-scale analysis [5, 6], Shells and Spheres preserves sharp boundaries with increasing scale.

This thesis develops the concepts of Shells and Spheres, VSS, and potential applications of both to image analysis, with an emphasis on segmentation. Utilizing this framework, a wide variety of diverse algorithms may be created for sphere map optimization, as well as subsequent image analysis tasks. The information contained in an optimized sphere map has applications to a number of such tasks in addition to segmentation, including registration, structure identification, and measurement.

To begin, Chapter 2 provides background information on related topics in image analysis. These areas of study represent classic and modern concepts and algorithms with which the Shells and Spheres framework shares common ground, either in operation or objectives.

Chapter 3 details the methodology of the Shells and Spheres framework, including the notation adopted for description of the framework and algorithms based on it. The general principles of VSS and the statistical calculations currently used for analysis are presented in detail, as well as a discussion of their specific performance compared to conventional image analysis techniques.

Utilizing the Shells and Spheres framework and VSS, a variety of algorithms can be developed to optimize a sphere map and subsequently perform other image analysis tasks. One such algorithm, designed to optimize the sphere map for an image and extended to identify medial and boundary locations is presented in Chapter 4. The medial and boundary information extracted is then applied to the task of image segmentation by utilizing a specially designed flood-fill operation.

Chapter 5 presents and discusses results of the application of the system to real images of the aortic arch. Segmentation results are presented for both 2D and 3D contrast-enhanced CT images of the aortic arch. Tests are conducted on synthetic images with added white noise to gauge performance of the algorithm in increasingly noisy conditions.

Finally, Chapter 6 will discuss various conclusions and proposes further research for the Shells and Spheres framework and VSS.

2.0 BACKGROUND

The essential purpose of the Shells and Spheres framework is to deduce the *scale* of each pixel in an image. Scale is well-established in image analysis as a fundamental concept. The analysis approach taken by the Shells and Spheres framework shares qualities with the widely used concept of “scale space” [5, 6], as well as other popular analysis techniques utilizing the principles of pixel scale, including Distance Maps [1], Fuzzy Connectedness [7], and Mathematical Morphology [8], although the approach is different from all of these. A discussion of the basic concepts of these techniques and the rationale for the divergence of Shells and Spheres is fitting at this juncture.

2.1 SCALE SPACE

The concept of *scale space*, as put forth by Lindeberg [5, 6], adds a dimension to an image by blurring it with a gaussian whose standard deviation increases along this new dimension. Thus, each pixel becomes a line along which it incorporates information from an even larger, spherically symmetrical region in the image surrounding it. However, with Shells and Spheres, for which the boundary of the spherical region is sharp, scale space inherently defines spherical regions whose boundaries taper asymptotically to zero, associating larger scales with blurred boundaries. As will be seen, Shells and Spheres avoids this unfortunate association.

2.2 DISTANCE MAPS

The calculation of distance maps in image analysis has been well studied. [1, 9, 10, 11, 12, 13, 14, 15] A distance map simply contains the distance from each pixel in an image to the nearest boundary. A correctly optimized *sphere map* produces, in essence, a distance map, as the radius of each sphere will be the distance to the boundary that caused the sphere to stop its growth.

The computation of classical distance maps requires the previous determination of boundaries within an image. Boundaries can be detected in many ways, the most common classical methods involving gradient calculations or intensity thresholding. Traditional gradient-based boundary calculations are generally problematic, as they are based on fixed-size kernels, which consider only small amounts of local image information when determining the location of boundaries. Noise, particularly non-uniform or spatially low-frequency noise, can generate non-boundary gradients capable of confusing gradient-based boundary detectors. Fixed-sized boundary kernels have the added limitation that boundary location within a given kernel placement is not reported, only the presence of the boundary somewhere within the kernel is known. Thresholding image intensity introduces a separate set of concerns. For ideal performance, homogeneity within objects, or at least low intra-object variance, is desired. This is rarely, if ever, the case in real images. Additionally, finding the boundaries of multiple objects of different intensities within the same image may require multiple thresholds, and intelligent determination of the number of thresholds and the optimum value of each is difficult to automate.

The system presented in this dissertation was designed to address the aforementioned shortcomings of classical boundary detection methods. To avoid the localization problem of fixed-size kernels, we utilize the specialized VSS gradient, which is detected over a range of sphere radii. This method allows verification of boundary gradients at multiple scales to minimize instances of misinterpretation of noise as a boundary. In addition, boundaries are subsequently adjusted and sharpened through the use of sets of spheres determined to be in separate objects, across a shared boundary from each other. Statistical population tests are used to determine the memberships of these sets, as well as the exact location of the

boundary between them. The power of this method of boundary determination comes from the ability of the sets to conform to arbitrary shapes and boundaries by combining spheres of various sizes within the same object and from the consideration of large populations of pixels on either side of a boundary in determining its exact location. By detecting each homogeneous region in the image and using their independent statistical values, the system overcomes the difficulties inherent to global thresholding by removing the need for multiple explicit thresholds and is able to behave intelligently and independently in each object within the image.

2.3 FUZZY-CONNECTEDNESS

In their recent papers on Fuzzy-Connectedness [7, 16, 17], Saha *et al.* describe a set of *hyperballs* centered at each image pixel, grown as large as possible without crossing boundaries. This is the same basic goal presented for the spheres in this dissertation. Hyperballs provide scale information for pixels to be used as a component of an affinity equation, which directs the fuzzy-connectedness algorithm. The method used for calculating the size of each of these hyperballs is detailed in [17]. The measure used to determine scale is a value denoted $FO_{k,v}(c)$, which is the “fraction of the object” of a hyperball $B_{k,v}(c)$ of radius k centered at pixel c . The value of $FO_{k,v}(c)$ is calculated using a few fundamental values and concepts. First, a user-defined homogeneity function W is used to find the probability (0-1) that two pixels are in the same homogeneous region, given the intensity values of these two pixels c and e , as in $W(|f(c) - f(e)|)$. In the calculation of $FO_{k,v}(c)$, this function is used to find the homogeneity measure between each pixel e in the outer shell of the hyperball, as represented in set notation by the subtraction of a sphere one size smaller from the current sphere $B_{k,v}(c) - B_{k-1,v}(c)$, and the central pixel c . Using these values, the equation for $FO_{k,v}(c)$ is given as

$$FO_{k,v}(c) = \frac{\sum_{e \in B_{k,v}(c) - B_{k-1,v}(c)} W(|f(c) - f(e)|)}{|B_{k,v}(c) - B_{k-1,v}(c)|}. \quad (2.1)$$

The parameter v is a resolution value, and can be ignored in this discussion. The numerator can be recognized as the sum of the homogeneity likelihood values given by W for each pixel e in the hyperball's outer shell when compared to the central pixel c . The denominator is a normalizing factor, representing the number of pixels contained in the outer shell being considered.

Thus, in a noiseless image, a sphere with an outer shell completely within a uniform intensity region would have an $FO_{k,v}(c)$ value of 1. A manual threshold t_s , under which the $FO_{k,v}(c)$ values of spheres must remain in order to continue growth, is required. Saha, et al. present a variety of possible homogeneity functions, each of which will perform differently on a given image.

This method of scale calculation is not designed for extreme accuracy, and thus provides a somewhat unreliable measure of scale. A number of issues with this approach become readily apparent:

- using only the outer shell greatly decreases the statistical population used, weakening the measurement
- the individual measure of homogeneity is derived from the comparison of only two pixels (c and e), rather than an entire population
- the necessity of choosing the homogeneity function W adds a knowledge-dependent parameter to the system, which must be rectified by an experienced user
- the entire approach is highly dependent on having either uniform noise, or a very complex homogeneity function designed to account for a specifically known type of non-uniform noise

In noisy images, these drawbacks become cause for concern. Using only the central pixel c as a basis for homogeneity comparisons is problematic, especially if that pixel is afflicted with a large amount of noise. Homogeneity functions must be chosen very carefully by a knowledgeable user. The interior region of the ball is not considered in any way, meaning that the number of pixels considered to make a decision is small, compared to the pixel population of the entire hyperball. There are two ways of considering this approach. Since the population being considered is small compared to the entire sphere, a few pixels outside

the region should have a more profound impact on the sum of the outer shell, and therefore the sphere growth decision, than the same few pixels would have had when averaged in to the pixel population of the entire sphere. This can be seen as a positive feature. However, giving more weight to a few pixels can also be detrimental, as a few noisy pixels may have a dramatic impact on the homogeneity value calculated. Viewed in this light, it can be seen that the smaller population is more susceptible to the negative effects of noise and other sources of inhomogeneity than a system using the largest population possible, which is the approach used in Shells and Spheres.

2.4 MATHEMATICAL MORPHOLOGY

Mathematical Morphology [18, 8] should be discussed, as some conceptual similarities may be seen between it and Shells and Spheres. The growing and shrinking actions of spheres in Shells and Spheres may be likened to erosion and dilation of shapes in Mathematical Morphology, and set theory is prevalent in both approaches. Homogeneous regions in Mathematical Morphology contort when eroded or dilated depending on the image data. Sets of spheres in a Shells and Spheres system are able to combine to produce similarly complex shapes, though they are made up of individual spheres whose only variable is radius. A prominent distinction can be seen in the use of shape priors to govern shape evolution. Mathematical Morphology uses a set called the “structuring element” to govern the erosion or dilation of its shapes. This shape prior must be carefully selected using knowledge of the data to be analyzed and the results desired. Shells and Spheres, on the other hand, currently uses no shape prior in its analysis.

The above differences lead to notably different uses for these systems. Mathematical Morphology is known for being able to remove incorrect or unwanted shapes (such as the unwanted shorts in circuit boards) because an equal number of erosions and dilations using the same structuring element can produce a shape different from the initial shape. With Shells and Spheres, growing and shrinking a sphere an equal number of times always results in the same final set of pixels, with the same resulting descriptive statistics. Any variability in

this process comes from the algorithm used, and the way in which sets of spheres are created and maintained. Additional constraints such as minimum sphere size could allow a Shells and Spheres system to function in the same way, removing small objects while maintaining larger ones. The introduction of shape priors into a Shells and Spheres system leads to a whole new realm of possible algorithms and applications. None of these constraints, however, are inherent to the framework or the approach itself.

This example is given as an illustration of the flexibility of the Shells and Spheres framework, and its adaptability, via algorithmic design, to a wide variety of common image analysis tasks. It is believed that the framework represents a novel, general toolkit for image analysis, applicable to a variety of known tasks and algorithms as well as new and innovative applications.

2.5 MEDIALNESS

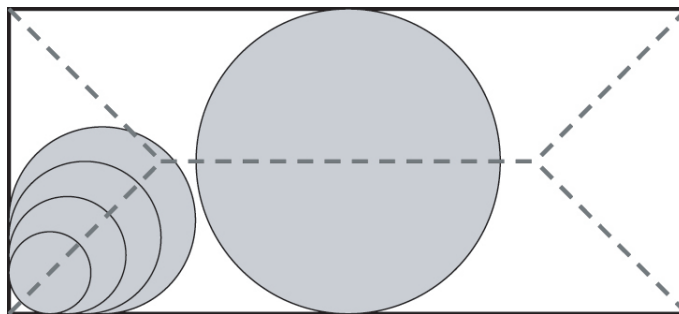


Figure 1: The Blum medial manifold (dotted line) of a rectangle, with selected medial spheres shown.

One other body of research needs to be included here, not necessarily for Shells and Spheres itself, but for the application to which it is applied in this dissertation. A fundamental goal of the analysis system presented is to determine and describe the shape of objects found within an image. While there are many possible descriptive methods, most are based on boundary detection and/or medial analysis. Boundary detection defines a shape

by finding points on the boundary of that shape. Medial analysis, instead, uses the points at the center of a given shape to define it. While both of these methods can accurately define shapes, the amount of data necessary for a medial representation is far less than for the boundary representation, making it a more efficient approach. Consider a sphere, which takes only one medial point (at the center) and a radius value to describe it. To describe this same sphere with a boundary representation, an ever-increasing number of boundary points, depending on the size of the sphere, must be used. Because of these innate advantages of medial points as shape descriptors, their determination has been assigned a pivotal role in the shape analysis methods presented in this thesis.

The classic medial manifold was defined by Blum [4]. This branching manifold, as seen in Fig. 1, can be intuitively understood as the locus of all points forming the centers of spheres (circles, in 2D) that can be placed entirely within an object and touch two or more separate boundaries. When touching a boundary, a sphere is in a state such that, were the sphere to grow any larger while maintaining its current center position, it would cross the boundary in question. This is true of all maximally-sized spheres inside an object, while medial spheres are distinctive in that they touch multiple boundaries, causing their centers to be located on the medial manifold of the object. This concept and definition of medial manifolds was actually inspirational in creating the Shells and Spheres approach, and finding medial points remains one of its primary applications.

3.0 METHODOLOGY

We will begin this section by defining notation. As previously mentioned, Shells and Spheres can be generalized to n -dimensions. The term *sphere* is used (instead of “circle” or “hypersphere”) regardless of the number of dimensions involved. Figures are presented in 2-dimensions for ease of illustration and explanation.

3.1 SHELLS AND SPHERES NOTATION

Since the framework of Shells and Spheres is used to gather statistics on dynamic collections of pixels, a hybrid form of notation is derived from standard set theory and statistics. Vectors are denoted by lowercase bold-faced letters (\mathbf{x}), scalars by lowercase italic letters (r), and sets by uppercase italic letters (S). \mathbb{Z} denotes the set of all integers, and $\Omega \subset \mathbb{Z}^n$ denotes the set of all pixel locations in a sampled n -dimensional image.

Given an n -dimensional image with intensities $f(\mathbf{x})$ for $\mathbf{x} \in \Omega$, we define a *radius image*, which assigns the radius $r(\mathbf{x})$ to the sphere centered at each pixel \mathbf{x} . The collection of all spheres in the image, with their individual radii defined by $r(\mathbf{x})$, is referred to as the *sphere map*.

A sphere¹ is defined as an n -dimensional neighborhood of pixels that lie within a radius r of a center point. Integer values are used for r , such that a sphere of radius r centered at

¹Here it should be noted that many mathematicians use “sphere” to mean the 2D outer surface of a spherical region and not the entire volume within, for which the term “ball” is reserved. We, however, will trace our use of the word to Webster’s Dictionary, and to its common meaning in physics and astronomy, where sphere means the entire volume within the spherical object.

a pixel \mathbf{x} is given by

$$S_r(\mathbf{x}) = \{\mathbf{y} : \text{round}(|\mathbf{y} - \mathbf{x}|) \leq r, \mathbf{y} \in \Omega\}. \quad (3.1)$$

Note the shorthand notation for the subscript r , meaning $r(\mathbf{x})$, the radius of the particular sphere at \mathbf{x} as given by the sphere map. In some instances, the reader will encounter an example with a different subscript, such as $S_1(\mathbf{x})$, meaning a sphere of radius 1, irrespective of $r(\mathbf{x})$. By definition, $\mathbf{x} \in S_r(\mathbf{x})$ for all \mathbf{x} , even when $r(\mathbf{x}) = 0$, and hence $S_r(\mathbf{x})$ is always non-empty.

A *shell* is a set of all pixels whose distance to the center rounds to a given radius, defined for radius r as

$$H_r(\mathbf{x}) = \{\mathbf{y} : \text{round}(|\mathbf{y} - \mathbf{x}|) = r, \mathbf{y} \in \Omega\}. \quad (3.2)$$

Shells are non-overlapping such that for concentric shells,

$$H_p(\mathbf{x}) \cap H_q(\mathbf{x}) = \emptyset, \quad p \neq q. \quad (3.3)$$

Additionally, shells are space-filling, and thus a sphere of radius r may be formed from a union of shells,

$$S_r(\mathbf{x}) = \bigcup_{k=0}^r H_k(\mathbf{x}). \quad (3.4)$$

Figure 2 illustrates the distribution of pixels in a series of concentric shells surrounding a central pixel in a 2D image. Each pixel is labeled with its offset from the central pixel (labeled “x”). Figure 3 shows an image containing two noiseless objects with pixel intensities of 1 and 9 respectively. Note that pixels in this case are represented by their intensity. The boundary between the objects is identified by a straight dashed line. Pixel \mathbf{x} is surrounded by a concentric set of four shells $H_0(\mathbf{x})$, $H_1(\mathbf{x})$, $H_2(\mathbf{x})$, and $H_3(\mathbf{x})$, shown separated by dashed circles. Shell $H_3(\mathbf{x})$ is truncated by the edge of the image. The union of all four shells is $S_r(\mathbf{x})$, shown enclosed by a solid circle, with a radius governed by the value of $r(\mathbf{x}) = 3$ in the sphere map. Similarly, on the other side of the boundary, pixel \mathbf{y} with a value of 9 has three shells whose union $S_r(\mathbf{y})$ has a radius $r(\mathbf{y}) = 2$. Both $S_r(\mathbf{x})$ and $S_r(\mathbf{y})$ touch but do not cross the boundary, and are therefore correctly optimized.

The correctly optimized sphere map of the image in Fig. 3 is shown in Fig. 4, with each pixel represented by the radius of the sphere centered at that pixel. Note the linear increase

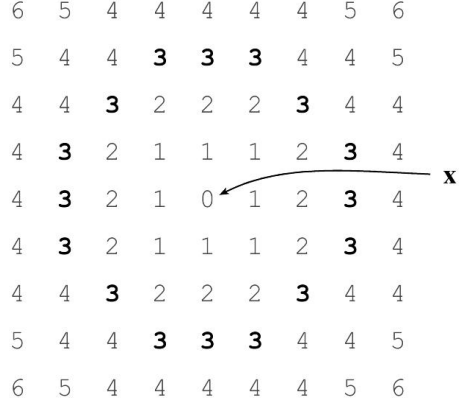


Figure 2: Each pixel is shown as a number indicating its integer distance from the central pixel. If we denote the central pixel as \mathbf{x} , then pixels labeled with integer n are members of the set $H_n(\mathbf{x})$. For example, the pixels labeled “3” (shown in bold) comprise the set $H_3(\mathbf{x})$.

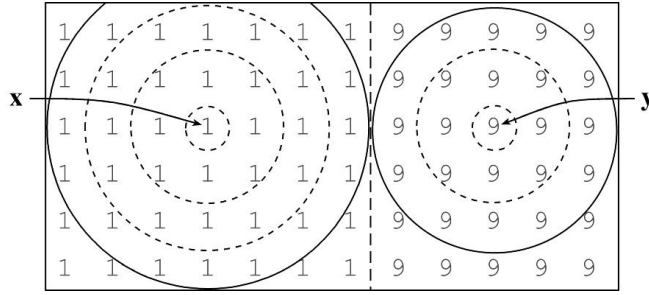


Figure 3: Noiseless image with boundary between two objects. Correctly scaled spheres $S_r(\mathbf{x})$ with $r(\mathbf{x}) = 3$ and $S_r(\mathbf{y})$ with $r(\mathbf{y}) = 2$ touch, but do not cross, the boundary. Numbers indicate pixel intensity.

in sphere radius with distance from the boundary and the fact that the radius equals zero adjacent to the boundary.

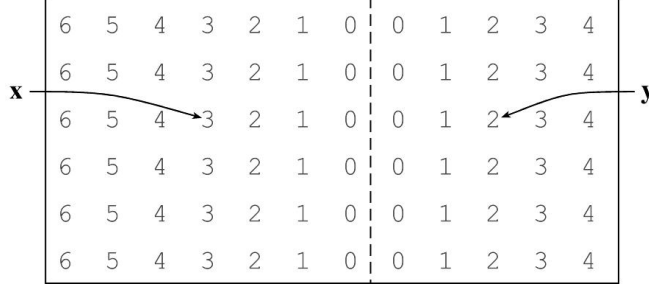


Figure 4: Correctly optimized sphere map of the image in Fig. 3. Numbers indicate the integer radius of the sphere at each pixel. Pixels \mathbf{x} and \mathbf{y} are labeled as before.

3.2 VARIABLE SCALE STATISTICS

A number of statistics are derived at each pixel \mathbf{x} , calculated from the intensities of pixels within various spheres within the sphere map. Since these statistics depend on the radii of the spheres, we call them Variable Scale Statistics (VSS). We denote as *primary statistics* those VSS at \mathbf{x} calculated using only the population of pixels within $S_r(\mathbf{x})$. Higher-order *secondary statistics* are VSS derived from multiple spheres.

3.2.1 Primary Statistics

The primary statistics at pixel \mathbf{x} concern only the population of pixels within the sphere $S_r(\mathbf{x})$. Thus the *mean* at pixel \mathbf{x} is the mean intensity of all pixels within the population $S_r(\mathbf{x})$, defined as

$$\mu(\mathbf{x}) = \frac{1}{|S_r(\mathbf{x})|} \sum_{\mathbf{y} \in S_r(\mathbf{x})} f(\mathbf{y}), \quad (3.5)$$

where $|S_r(\mathbf{x})|$ is the number of pixels in $S_r(\mathbf{x})$ and $f(\mathbf{y})$ is the image intensity at \mathbf{y} . The *variance* at pixel \mathbf{x} is defined as

$$\sigma^2(\mathbf{x}) = \frac{1}{|S_r(\mathbf{x})| - 1} \sum_{\mathbf{y} \in S_r(\mathbf{x})} [f(\mathbf{y}) - \mu(\mathbf{x})]^2. \quad (3.6)$$

where $|S_r(\mathbf{x})| > 1$. The standard deviation $\sigma(\mathbf{x})$ is simply the square root of the variance.

The *first-order moment* of intensities within $S_r(\mathbf{x})$ is given by

$$\mathbf{m}(\mathbf{x}) = \sum_{\mathbf{y} \in S_r(\mathbf{x})} (\mathbf{y} - \mathbf{x})f(\mathbf{y}). \quad (3.7)$$

Due to the finite extent of an image's domain Ω , a sphere may be truncated by one or more edges of the image (for example, $S_3(\mathbf{x})$ in Fig. 3). Unlike conventional kernels, which usually require pixel values to be arbitrarily defined outside the image, these spherical sets simply exclude such locations from all calculations. This truncation will not adversely affect $\mu(\mathbf{x})$ or $\sigma(\mathbf{x})$. However, the first-order moment will suffer a bias due to asymmetrical pixel distribution within the truncated sphere. To compensate for this bias, a measure that shows no edge effect is defined, called *VSS gradient*. Given the center of mass of pixel locations in sphere $S_r(\mathbf{x})$,

$$\mathbf{c}(\mathbf{x}) = \frac{1}{|S_r(\mathbf{x})|} \sum_{\mathbf{y} \in S_r(\mathbf{x})} \mathbf{y}, \quad (3.8)$$

the VSS gradient at \mathbf{x} is

$$\nabla f(\mathbf{x}) = \frac{1}{|S_r(\mathbf{x})|} [\mathbf{m}(\mathbf{x}) - \mu(\mathbf{x})(\mathbf{c}(\mathbf{x}) - \mathbf{x})]. \quad (3.9)$$

The VSS gradient does not suffer from the usual edge effects of convolution kernels (see Fig. 7C). Note that for non-truncated spheres $\mathbf{c}(\mathbf{x}) = \mathbf{x}$ and VSS gradient is equivalent to the moment vector normalized to the number of pixels, $\mathbf{m}(\mathbf{x}) / |S_r(\mathbf{x})|$.

All of the above statistics can be computed incrementally as shells are added to, or removed from, a given sphere, significantly reducing computational load during sphere map optimization.

3.2.2 Secondary Statistics

Secondary statistics are defined as higher order VSS derived by combining multiple spheres to form more complex neighborhoods. One such neighborhood, $S^{-1}(\mathbf{x})$, is defined as

$$S^{-1}(\mathbf{x}) = \{\mathbf{y} : \mathbf{x} \in S_r(\mathbf{y})\}, \quad (3.10)$$

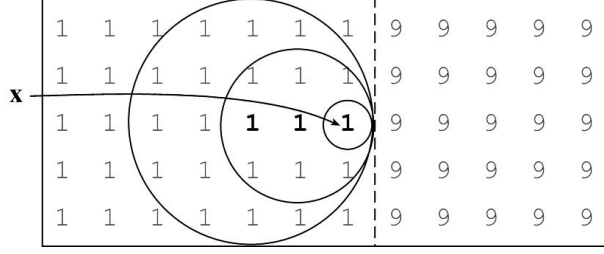


Figure 5: Set $S^{-1}(\mathbf{x})$ of spheres that contain pixel \mathbf{x} , adjacent to the boundary between two noiseless objects with respective intensities of 1 and 9. Numbers indicate the intensity value of the pixel. (centers of the spheres shown in bold)

the set of all pixels whose spheres contain \mathbf{x} . The -1 superscript is used to impart the flavor of an inverse function. Note that since it is always true that $\mathbf{x} \in S_r(\mathbf{x})$ it must likewise always be true that $\mathbf{x} \in S^{-1}(\mathbf{x})$.

Given an optimized radius image, $S^{-1}(\mathbf{x})$ will consist entirely of pixels within the same object as pixel \mathbf{x} . Fig. 5 shows members of one such $S^{-1}(\mathbf{x})$ set, consisting of three pixels (bold), whose spheres contain \mathbf{x} . Notice that all three spheres touch but do not cross the boundary, so this particular $S^{-1}(\mathbf{x})$ set is correctly optimized.

Secondary statistics are derived from populations of spheres such as $S^{-1}(\mathbf{x})$. Thus, the *mean of means* is defined as

$$\mu_\mu(\mathbf{x}) = \frac{1}{|S^{-1}(\mathbf{x})|} \sum_{\mathbf{y} \in S^{-1}(\mathbf{x})} \mu(\mathbf{y}), \quad (3.11)$$

or the mean of the mean intensities for all the spheres in $S^{-1}(\mathbf{x})$. In a noiseless image containing distinct homogeneous regions, $\mu_\mu(\mathbf{x})$ yielded by a correct sphere map will be identical to the original image.

Likewise, the *standard deviation of the means* is defined as

$$\sigma_\mu(\mathbf{x}) = \left[\frac{1}{|S^{-1}(\mathbf{x})| - 1} \sum_{\mathbf{y} \in S^{-1}(\mathbf{x})} [\mu(\mathbf{y}) - \mu_\mu(\mathbf{x})]^2 \right]^{\frac{1}{2}}. \quad (3.12)$$

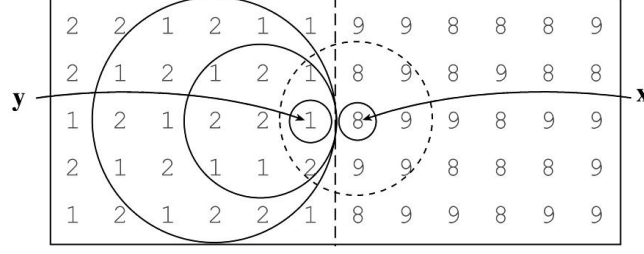


Figure 6: Image with noise. Numbers indicate the intensity value of the pixel. Pixel \mathbf{x} is deterred from extending its sphere across the boundary because its mean is an outlier in the population $S^{-1}(\mathbf{y})$.

Note that the above definition of $\sigma_\mu(\mathbf{x})$ is given only for $|S^{-1}(\mathbf{x})| > 1$. For noiseless images, a correct sphere map will yield $\sigma_\mu(\mathbf{x})$ values of 0 for all pixels. In an image containing noise, $\sigma_\mu(\mathbf{x})$ of spheres within the same region can provide a measure of the total inhomogeneity of the region. Note that this measure includes both local and global sources of inhomogeneity, such as tissue inhomogeneity, uniform image noise, scanner noise, etc.

The *z-value* of a pair of spheres is defined to provide a measure of how well $S(\mathbf{x})$ fits into the current $S^{-1}(\mathbf{y})$ set,

$$z_\mu(\mathbf{x}|\mathbf{y}) = \frac{|\mu(\mathbf{x}) - \mu_\mu(\mathbf{y})|}{\sigma_\mu(\mathbf{y})}. \quad (3.13)$$

The justification is that, in an optimized radius image, if $S_r(\mathbf{x})$ were to contain pixel \mathbf{y} , then $\mu(\mathbf{x})$ should fall well within the distribution of means for all spheres that already contain \mathbf{y} . This concept is illustrated in Fig. 6, which shows pixel \mathbf{x} attempting to extend its sphere across the boundary to include pixel \mathbf{y} . Noise has been included in the image to demonstrate that a high *z-value* could be used to stop the growth of $S_r(\mathbf{x})$ at the boundary, even in the presence of noise. It should be noted that the utility of this statistic is dependent on a reasonable initialization of the sphere map, such that the percentage of spheres not crossing boundaries is high enough to lend statistical validity to $\mu_\mu(\mathbf{y})$ and $\sigma_\mu(\mathbf{y})$.

3.2.3 Demonstration on a Noiseless Image

The optimization of a sphere map using the presented system is demonstrated on a noiseless synthetic image containing a number of homogeneous objects as shown in Fig. 7A. Optimization on such an image is trivial; spheres are initialized to zero radius and allowed to grow until adding the next shell would yield a non-zero variance $\sigma^2(x)$. Figure 7B shows results of optimizing the sphere map on the synthetic image in Fig. 7A. Figure 7B shows negative ridges (dark) along boundaries, and positive ridges (bright) along medial manifolds. Figures 7C and 7D show, respectively, the y component of the VSS gradient and the variance calculated using spheres of radius 5 at all pixels in the image. Note that neither the gradient nor the variance suffer from edge effects. Note also that Figs. 7C and 7D use a non-optimized sphere map, which includes spheres that cross boundaries. On this “perfect” image with completely uniform object intensity, both VSS gradient and variance would be zero for the optimized sphere map shown, where spheres all contain only homogeneous sets of pixels.

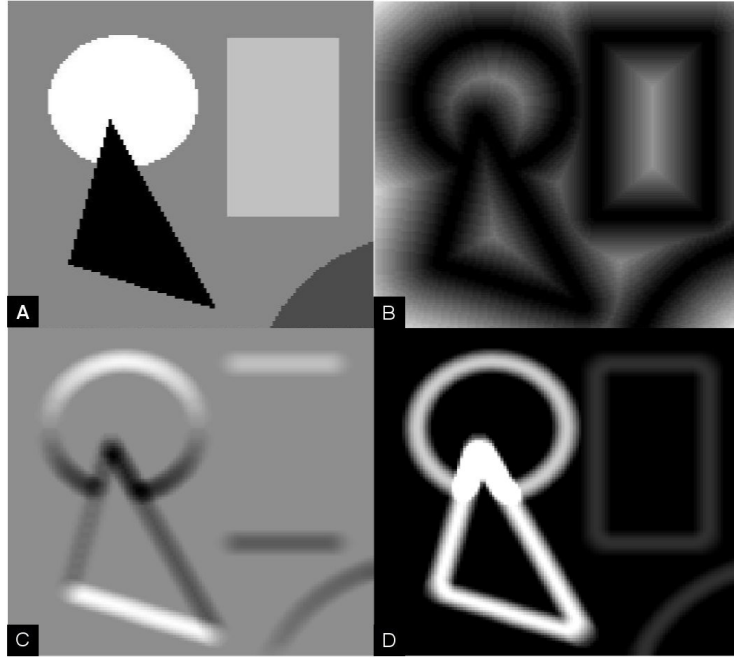


Figure 7: A: A noiseless synthetic image. B: The optimal radius image. C: The VSS gradient in the y direction, calculated using spheres of radius 5 pixels at all image points. D: The variance image, also calculated with all spheres set to a radius of 5.

4.0 ALGORITHM FOR SEGMENTING REAL IMAGES

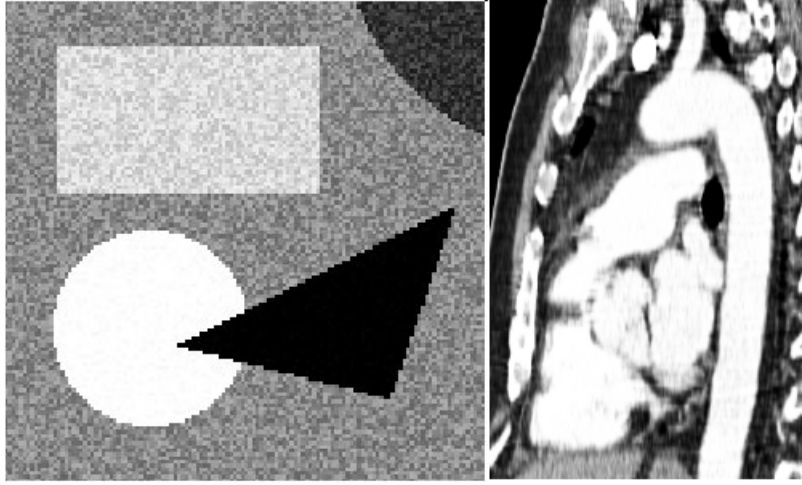


Figure 8: Images used for illustrative testing. Left: A 2D synthetic image with homogeneous regions, including added synthetic uniform noise (calculated using Equation 5.1, detailed in the Results chapter). Right: A contrast-enhanced CT scan of the aortic arch.

The Shells and Spheres framework presented thus far provides a basic set of versatile image analysis tools. A number of methods for the use of these tools have been explored, and in the remainder of this thesis one effective multistage algorithm for optimization of a sphere map will be presented. Along with its incremental performance segmentation of image objects in both real and synthetic data will be demonstrated. The presentation of this algorithm provides not only an example of a functional Shells and Spheres system, but also provides insight into the advantages, limitations, and overall utility of the Shells and Spheres framework.

The algorithm described takes the form of a six step process, with steps 1-4 optimizing

the sphere map, step 5 finding medial pixels, and step 6 producing an object segmentation. Step 1 creates an initial approximation of the sphere map, from which acceptable statistical values can be obtained for use in subsequent steps. Step 2 utilizes the pronounced discrepancy in variance between spheres that have incorrectly grown across boundaries and those that have not to reduce the size of incorrect spheres, placing them correctly within their appropriate image objects. Step 3 introduces specialized boundary indicators, known as *outposts*, to the image via boundary information extracted from population testing between spheres in adjacent image objects. These outposts then influence the radii of nearby spheres, resizing them to adhere to the established boundary information. Step 4 revisits variance calculation using the current sphere map, which is more accurate than the sphere map previously available. The new variance measure is applied to spheres, encouraging them to fully grow within their respective image objects, effectively smoothing the radius image and sharpening its “boundaries” (as defined by all spheres that remain at the initial scale of 0, or a size of 1 pixel) simultaneously. Step 5 identifies medial pixels, i.e. those whose spheres touch at least two boundaries. Given a medial seed point, step 6 connects neighboring medial pixels and combines their corresponding spheres to produce a segmentation. The following sections describe each step in more detail.

To illustrate the effect of each step more clearly, the progress of the radius image will be followed on a synthetic image with added noise and a contrast-enhanced CT scan of the aortic arch, both shown in Fig. 8. To clearly display the performance of each component of the algorithm, results will be presented after each phase of development of the radius image, i.e. after each of the six steps presented, for both images. Stepwise results are presented on 2D images only, for ease of visual presentation, though final results will also be presented for the full 3D aortic arch data set. The 3D results presented are not compiled from a series of 2D data slices. Rather, they are intrinsic 3D calculations, as the algorithm presented, like the Shells and Spheres framework itself, is inherently n-dimensional.

4.1 STEP 1: VSS GRADIENT-BASED RADIUS APPROXIMATION

For the noiseless synthetic image shown earlier (Fig. 7), it is trivial to optimize the sphere map by growing the spheres until any non-zero variance is detected. When analyzing real images, however, this approach will fail, because intensity variation due to noise may be indistinguishable from an object boundary, especially within small spheres. To yield viable statistical populations for region representation, at least some spheres must be correctly initialized. To accomplish this, all spheres are first set to $r(x) = 0$ and are then allowed to grow until a persistent increase in VSS gradient magnitude is detected over a series of scales. Unlike conventional gradient measured with a fixed-scale kernel, the VSS gradient depends locally on $r(\mathbf{x})$, and is based on the first-order moment of intensity normalized by the number of pixels in the sphere. Thus the VSS gradient can be expected to increase monotonically as a sphere grows past a boundary, since the first order moment favors the outer pixels. A persistent increase in VSS gradient for g consecutive steps is sought, at which point $r(\mathbf{x})$ is reset to the scale just before the increase began. Empirically, it has been found that a value of $g = 5$ performs well for the data tested to achieve a reasonable first approximation of the optimized sphere map. Theoretically, the ideal value of this parameter would change in proportion to the amount of uniform image noise, as a higher level of noise would cause more obfuscation of boundaries, impeding the detection of gradient. Note that any increase in gradient, regardless of magnitude, will count as one of the g increases. In the same manner, even the slightest decrease in gradient will break the chain of consecutive gradient increases being sought. Noise can cause random small fluctuations in gradient when no actual gradient is present. Because of the possibility of this anomaly occurring just before a growing sphere encounters an actual gradient, it is not unlikely that a sphere’s position at the end of Step 1 will be one or two steps short of the actual boundary. A less likely, but still possible, error is the case in which noise may cause enough of a decrease in perceived gradient to cause a break in the chain of g consecutive gradient increases caused by an actual boundary. Since it is far more likely that noise could overcome the influence of an actual boundary on the VSS gradient within a small population of pixels (i.e. a small-scale sphere), it is equally likely that such an error would only occur early in a sphere’s growth, again only

deviating the sphere's end position by a few units of radius from the ideal.

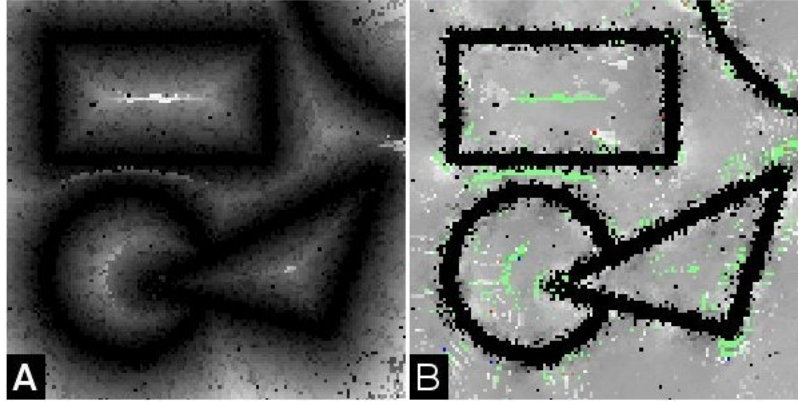


Figure 9: A: The sphere map $r(\mathbf{x})$ of the image after Step 1. B: The variance of each pixel after Step 1. The grayscale range of the image is 0-300, while all green pixels are distinctly higher than this range.

Figure 9A shows the state of the sphere map after this first step for a the noisy synthetic image presented, while Fig. 11A shows the same information for a contrast-enhanced CT scan of the aortic arch. When examining the $r(\mathbf{x})$ image in Fig. 9A, two problems are immediately apparent. First, there is sporadic noise in the scales of the pixels, leaving them often with lower intensity than they should be, indicating spheres that have not grown enough. These spheres have been stopped early by noise that was perceived as boundary. Second, there are abnormally high scale pixels in the centers of objects. These values represent spheres that have grown past two symmetrical boundaries on opposite sides of the sphere because the gradient contributions of the opposing boundaries canceled each other, producing an incorrectly small moment (See Eqn. 3.7). As will be seen in the following section, this latter problem can be addressed using variance.

This first step is effective at growing spheres past tissue inhomogeneity and noise. However, spheres may not stop exactly on the boundary because of the aforementioned effect of noise on the detection of gradient increase. Using VSS gradient to govern sphere growth can also fail completely for a sphere that encounters two opposing boundaries simultaneously, as their contributions to the gradient may cancel.

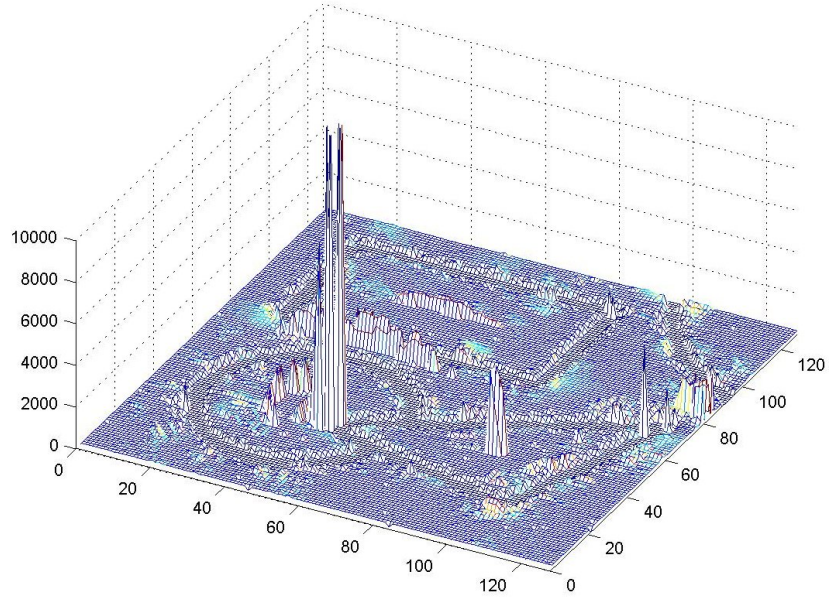


Figure 10: A colored height map of the variance image from Fig. 9B

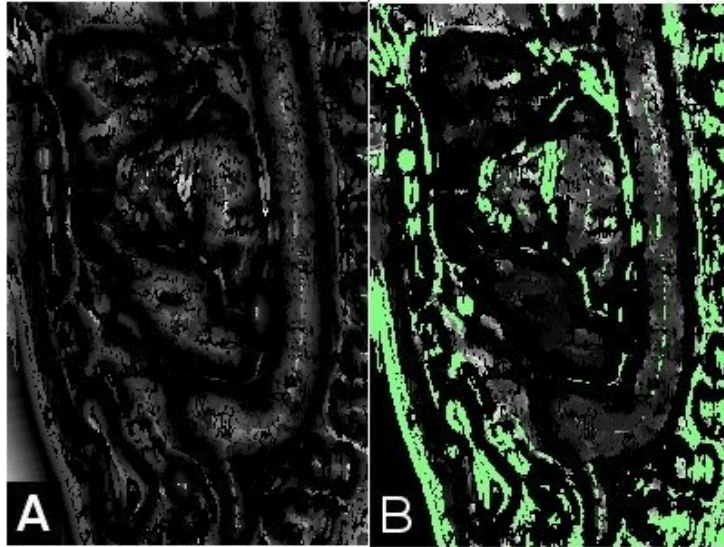


Figure 11: A: Sphere map $r(\mathbf{x})$ after Step 1. B: The variance of each pixel after Step 1. The grayscale range of the image is 0-915, while all green pixels are distinctly higher than this range.

4.2 STEP 2: VARIANCE-CONSTRAINED RADIUS REDUCTION

After Step 1, three possible states exist for each sphere: The sphere can be too large, too small, or the correct size (i.e. it touches the nearest boundary but does not cross it). The most glaring error in the sphere map after Step 1 is the presence of large-scale spheres that have incorrectly grown past boundaries. As previously mentioned, this type of error typically occurs when a growing sphere contacts multiple boundaries at once, which in many cases indicates that the sphere lies on the medial manifold. In such a case, the contributions to the VSS gradient from multiple boundaries may cancel, allowing the sphere to grow larger than its correct radius.

Examples of such erroneously high-variance pixels can be seen in Fig. 9B. These incorrect pixels are colored green because their variance value is exponentially higher than the grayscale pixels. While the grayscale portions of the image have a range of intensity values from 0 to 300, the green values increase to nearly 16000. The full range of the variance of these improperly grown pixels can be easily seen in Fig. 10. The flat “floor” section of the image is the average variance produced by the noise added to the image, which has a value near 200. The spheres near boundaries are seen to have distinctly lower variance, since they are very small scale, and thus contain very few pixels. Figure 11 shows the same data described above for a contrast-enhanced CT scan of the aortic arch. The same problematic trends in the sphere map can be seen, as well as similar high variance issues caused by spheres growing past boundaries. Note that the range on the grayscale areas in the variance image (Fig. 11B) is 0-915, differing from that of its synthetic counterpart.

These erroneous spheres will, however, have a larger variance $\sigma^2(\mathbf{x})$ than spheres that correctly remain within the object boundaries, a fact which can be exploited to correct them. To address the problem, all spheres with a variance above a certain threshold α are shrunk by decrementing $r(\mathbf{x})$ for each sphere $S_r(\mathbf{x})$ until $\sigma^2(\mathbf{x}) < \alpha$. The threshold α is set relative to μ_{σ^2} and σ_{σ^2} , the mean and standard deviation, respectively, of the variance of all the spheres in the current sphere map, as defined by

$$\alpha = \mu_{\sigma^2} + \beta\sigma_{\sigma^2}. \quad (4.1)$$

The positive constant β represents the number of standard deviations above the mean permitted for a sphere's variance without the sphere being required to shrink. This parameter is not particularly sensitive, as the difference between spheres with erroneously high variance and all others is approximately two orders of magnitude. For the results presented, a value of $\beta = 0.2$ was used.

An arbitrary global threshold for $\sigma^2(\mathbf{x})$ is not ideal, because it assumes a constant expected variance throughout the image. This expectation is likely untrue, given factors such as tissue inhomogeneity and nonuniform noise, and future work may explore locally normalized thresholds. It should also be noted that the threshold depends on the current $r(\mathbf{x})$, which is not yet fully optimized at this step. This deficiency is addressed by returning to variance in Step 4, once a more accurate $r(\mathbf{x})$ is available.

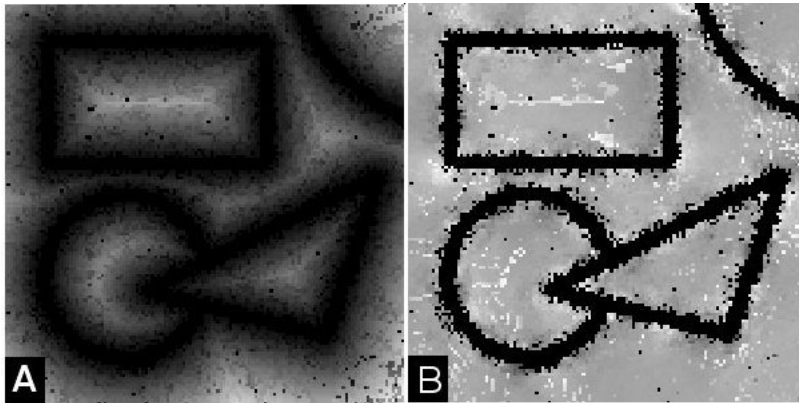


Figure 12: A: The sphere map $r(\mathbf{x})$ of the image after Step 2. B: Variance image after Step 2. The grayscale range of the image is 0-300.

Reducing the radius of pixels with extremely high variance will correct a majority of the spheres that have incorrectly grown past boundaries. The updated variance height map after Step 2 is shown in Figure 13, and effectively illustrates the fact that the high variance regions in the centers of objects have been eliminated.

This effect can also be seen in the variance and radius images (Fig. 12A and Fig. 14B, where the abruptly large-scale pixels at the centers of objects have been corrected.

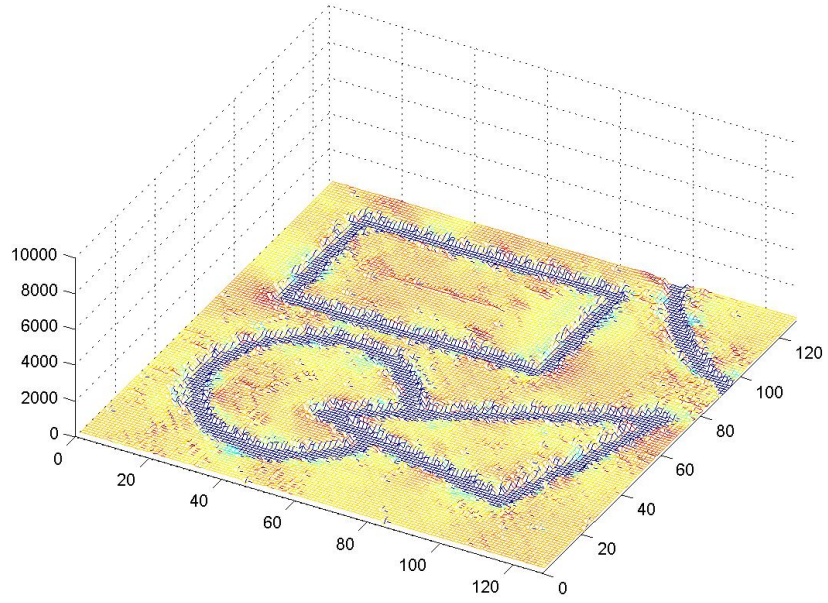


Figure 13: A colored height map of the variance image from Fig. 12B, showing the state of variance after Step 2.

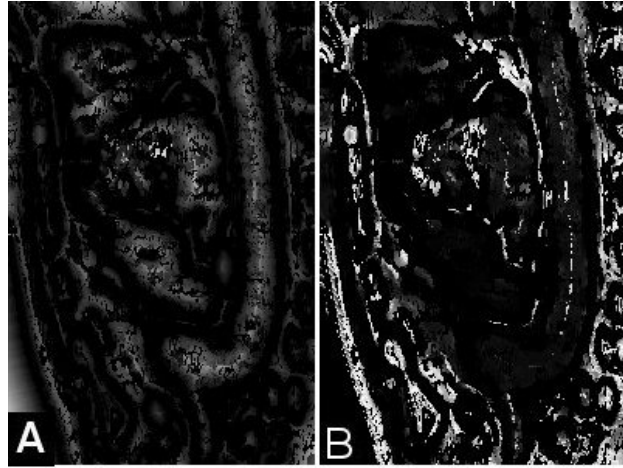


Figure 14: A: Sphere map $r(\mathbf{x})$ after Step 2. B: An image of the variance of each pixel after Step 2. The grayscale range of the image is 0-915.

4.3 STEP 3: OUTPOST SELECTION AND EXCLUSION

A primary use of VSS is to differentiate regions on opposite sides of the boundary. To do this, however, the radius image $r(\mathbf{x})$ must be in a state such that most spheres are not crossing boundaries, as spheres containing pixels from multiple image objects will decrease the effectiveness of statistics by representing samples from more than one object. Following Step 2, it is generally the case that enough spheres are correctly sized that many face each other across boundaries, producing high values for $z_\mu(\mathbf{x}|\mathbf{y})$. Likely boundary candidates are identified for a given sphere by finding a pixel \mathbf{y} in its $H_{r+1}(\mathbf{x})$ shell with a high value for $z_\mu(\mathbf{x}|\mathbf{y})$. The sphere at \mathbf{x} is said to place a *reflector* at such a location, a metaphorical construct denoting a vote by the sphere for the pixel \mathbf{y} as a point across its nearest boundary. Note that we do not vote for this pixel as a boundary itself, since boundaries may exist between pixels, as in Figure 3. Instead, pixels on each side of a detected boundary are marked. Thus, referring to Fig. 6, $S_r(\mathbf{x})$ could place a reflector at pixel \mathbf{y} . The set of reflectors placed by a given sphere $S_r(\mathbf{x})$ is denoted $K(\mathbf{x})$. In the present algorithm the constraint $|K(\mathbf{x})| = 1$ is applied, limiting each sphere by allowing it to place only one reflector for reasons discussed below. This constraint leads to the definition of $K(\mathbf{x})$ as

$$K(\mathbf{x}) = \{\mathbf{y} : \mathbf{y} = \underset{\mathbf{y} \in H_{r+1}(\mathbf{x})}{\operatorname{argmax}} z_\mu(\mathbf{x}|\mathbf{y})\}. \quad (4.2)$$

Such a constraint allows for the use of an **argmax** in determining the placement of a sphere's reflector, rather than requiring a decision of which pixels (and their corresponding $z_\mu(\mathbf{x}|\mathbf{y})$ values) in its $H_{r+1}(\mathbf{x})$ may represent a boundary and which do not. If a boundary exists just beyond the outer shell of $S_r(\mathbf{x})$, it will be located at the pixel \mathbf{y} for which the highest $z_\mu(\mathbf{x}|\mathbf{y})$ is calculated.

Each pixel may contain reflectors placed by a number of spheres. The set of spheres that have placed reflectors at \mathbf{x} is defined as

$$K^{-1}(\mathbf{x}) = \{\mathbf{y} : \mathbf{x} \in K(\mathbf{y})\}, \quad (4.3)$$

invoking the same inverse notation used for $S^{-1}(\mathbf{x})$ in Eq. 3.10. Figure 15 shows a set of spheres placing their reflectors across a boundary at pixel \mathbf{x} .

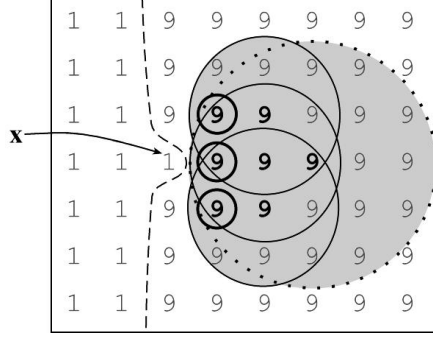


Figure 15: Illustration of $K^{-1}(\mathbf{x})$ containing 7 pixels (bold), each of whose sphere would place its reflector across the boundary at \mathbf{x} .

$|K^{-1}(\mathbf{x})|$ is referred to as the *reflector count*, i.e. the number of reflectors that have been placed at location \mathbf{x} . For example, in Figure 15, the reflector count $|K^{-1}(\mathbf{x})| = 7$.

A reflector placed by \mathbf{y} at \mathbf{x} has an inherent direction governed by the vector $(\mathbf{y} - \mathbf{x})$. The vector sum of the directions of all of the reflectors at \mathbf{x} is denoted the *reflectance* $\mathbf{k}(\mathbf{x})$, defined by

$$\mathbf{k}(\mathbf{x}) = \frac{1}{|K^{-1}(\mathbf{x})|} \sum_{\mathbf{y} \in K^{-1}(\mathbf{x})} \frac{\mathbf{y} - \mathbf{x}}{|\mathbf{y} - \mathbf{x}|}. \quad (4.4)$$

This measure provides the average orientation of the $K^{-1}(\mathbf{x})$ population, which describes the direction normal to the boundary, pointing to the center of the region represented by $K^{-1}(\mathbf{x})$. Since it was decided that each sphere will contribute exactly one reflector, reflector density and reflectance are normalized over the image. Therefore, reflector count can be used to differentiate between significant collections of reflectors correctly placed at boundaries and sparse distributions of reflectors incorrectly placed in the interior of objects. To denote pixels containing a significant number of reflectors, the term *outpost* is adopted, since such pixels serve as border markers, and in fact face each other across boundaries much the same way that military outposts of opposing armies face each other across the battle line.

The set of all pixels in an image chosen to be outposts is denoted by P . In the present algorithm this set is found in two steps. First, the set of *primary outposts* P' is established,

containing all pixels with zero radius and at least κ reflectors, i.e.

$$P' = \{\mathbf{x} : |K^{-1}(\mathbf{x})| \geq \kappa, r(\mathbf{x}) = 0\}. \quad (4.5)$$

For the results presented in this thesis, $\kappa = 4$. To increase the density of outposts along the boundaries, a set of *secondary outposts* P'' is generated, containing all pixels with zero radius that adjoin an outpost in P' and have at least λ reflectors, where $\lambda < \kappa$,

$$P'' = \{\mathbf{x} : |K^{-1}(\mathbf{x})| \geq \lambda, H_1(\mathbf{x}) \cap P' \neq \emptyset, r(\mathbf{x}) = 0\}. \quad (4.6)$$

For the results presented, $\lambda = 2$. By combining the sets of primary and secondary outposts, the set of all outposts,

$$P = P' \cup P'' \quad (4.7)$$

is formed.

Each outpost $\mathbf{y} \in P$ has a reflectance $\mathbf{k}(\mathbf{y})$. A sphere at \mathbf{x} can distinguish whether a given outpost is on its side of the boundary, constituting a *friendly outpost*, or the other side of the boundary, constituting an *enemy outpost*, based on the direction of the outpost's reflectance. The set of enemy outposts (those with reflectance facing \mathbf{x}) within the sphere of radius $r(\mathbf{x})$, is defined as

$$E_r(\mathbf{x}) = \{\mathbf{y} : \mathbf{y} \in P \cap S_r(\mathbf{x}), \mathbf{k}(\mathbf{y}) \cdot (\mathbf{y} - \mathbf{x}) < 0\}, \quad (4.8)$$

where the sign of the dot product determines the direction of $\mathbf{k}(\mathbf{y})$ relative to \mathbf{x} , differentiating friendly from enemy outposts. The exceedingly unlikely case of an outposts with $\mathbf{k}(\mathbf{y}) = 0$ can be considered a friendly outpost, as it does not indicate a boundary facing the sphere.

In governing the growth of a sphere, enemy outposts are to be avoided, while friendly outposts can be included. More specifically, enemy outposts stop the growth of spheres, as they represent a different image region than the one in which the sphere resides, while friendly outposts do not. Step 3 uses the number of enemy outposts to adjust the sphere size as follows: If the pixel contains no enemy outposts in $S_{r+1}(\mathbf{x})$, the sphere grows until it does. That is

$$\text{If } |E_{r+1}(\mathbf{x})| = 0, \text{ increase } r(\mathbf{x}) \text{ until } |E_{r+1}(\mathbf{x})| > 0.$$

If the number of enemy outposts in $S_r(\mathbf{x})$ is greater than γ , the radius is decreased until this is no longer true, i.e.

$$\text{If } |E_r(\mathbf{x})| > \gamma, \text{ reduce } r(\mathbf{x}) \text{ until } |E_r(\mathbf{x})| \leq \gamma.$$

In the present implementation, $\gamma = 2$. This factor prevents lone pixels that have been improperly labeled as outposts from incorrectly causing spheres to shrink.

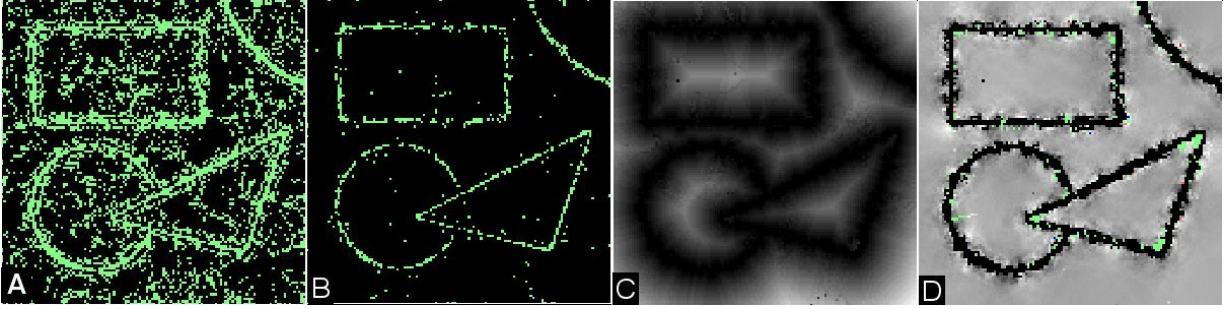


Figure 16: A: Reflector count of all pixels after Step 3 with a binary threshold ($|K^{-1}(\mathbf{x})| \geq 1$). B: Reflector count of all pixels after Step 3 with $|K^{-1}(\mathbf{x})| \geq 4$. C: The radius image $r(\mathbf{x})$ after Step 3. D: Variance after Step 3, with a grayscale range of 0-300.

After Step 2, the most pressing issue with the sphere map is the scattered effects of noise on $r(\mathbf{x})$, leading to the noise seen in the radius image in Fig. 12A. Now that the incorrectly large spheres have been adjusted to a more correct size, the next logical step is to focus on the small-scale pixels that incorrectly stopped growth at image noise, instead of growing to an actual boundary. Outposts provide a means for correcting these errors, as well as for adjusting spheres that have grown slightly too large or too small relative to their nearest boundary. The effect of these outpost-driven operations is that significant densities of reflectors along boundaries are used to correctly govern the size of spheres, sweeping incorrect reflectors from the centers of objects to the boundaries, as a sphere's reflector is redistributed when its radius is altered. Because most spheres are large relative to the spacing of outposts

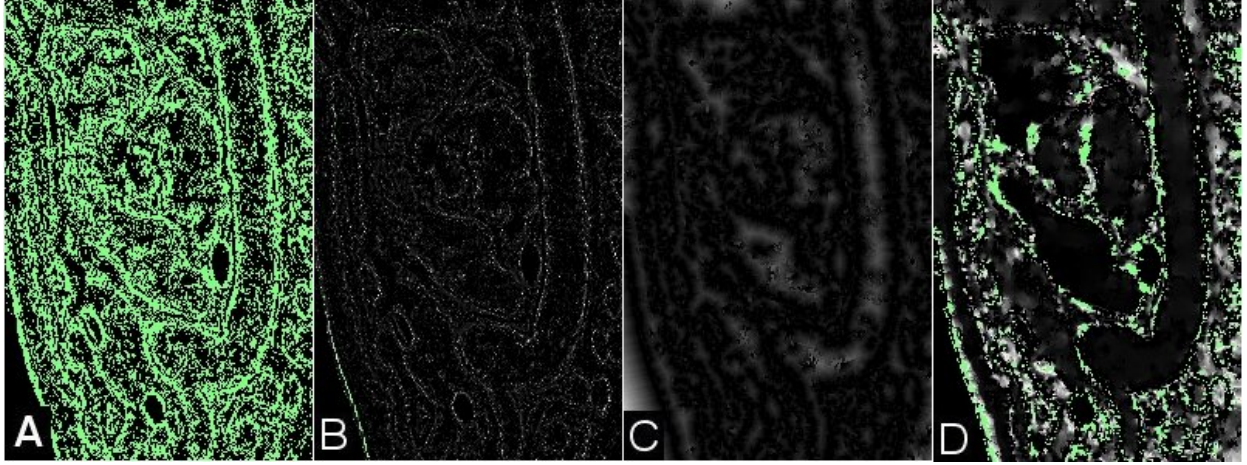


Figure 17: A: Reflector count of all pixels after Step 3 with a binary threshold ($|K^{-1}(\mathbf{x})| \geq 1$). B: Reflector count of all pixels after Step 3 with $|K^{-1}(\mathbf{x})| \geq 4$. C: The radius image $r(\mathbf{x})$ after Step 3. D: Variance after Step 3, with a grayscale range of 0-915.

along the boundary, their growth will be stopped and they will not “leak” or “bleed” across boundaries.

Figure 16 shows the state of the system after outpost selection and exclusion, adding reflector output to the previous figure format. Figure 16A is a binary thresholded image showing all pixels that have had at least one reflector placed at them. This set of pixels is dense, and while the true boundary pixels can be seen, a large number of false boundaries are also represented. Recall that only pixels with values of $|K^{-1}(\mathbf{x})| \geq 4$ are selected as primary outposts, provided they also have a radius value of $r(\mathbf{x}) = 0$. These candidate outpost pixels are shown in Fig. 16B. This set of pixels is far more representative of the actual boundaries in the image, yielding an effective basis for sphere radius correction.

Figure 18 shows the outposts calculated for each of the test images presented. Red pixels indicate primary outposts, and yellow pixels indicate secondary outposts. Comparing these images with the data shown in Figures 16B and 17B allows us to see the effects of creating outposts from reflectors, and the way in which the boundary information is enhanced by adding the radius constraint to the reflector thresholds κ and λ .

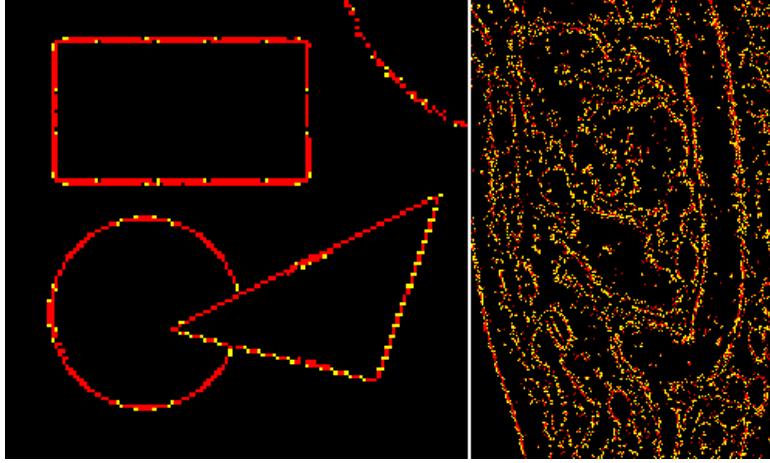


Figure 18: Left: Primary (red) and Secondary (yellow) outposts calculated for the synthetic test image. Right: Primary (red) and Secondary (yellow) outposts calculated for the real test image.

As can be seen in Fig. 16C, much of the sporadic noise in $r(\mathbf{x})$ has been removed by Step 3, leaving a noticeably smoother sphere map. This step also has a notable effect on the variance, shown in Fig. 16D. As spheres cease crossing boundaries, variance inside objects becomes much more uniform, as can be expected (compare Fig. 16D to Fig. 12B). It should also be noted that this step yields a much more representative image variance than the variance at the onset of Step 2, used to calculate the threshold α (Eqn. 4.2).

Figure 17 shows the same results for the real CT image, and the same basic elements of progress can be seen. For this real image, one will note that the reflector count images are less concentrated at actual boundaries, and the sphere map is far more inconsistent than its synthetic counterpart. However, even this seemingly sparse set of correct outposts is effective in correcting a large percentage of the sphere radii, as can be seen in the increased smoothness in the sphere map between Figure 14B and Figure 17C.

4.4 STEP 4: VARIANCE-CONSTRAINED SCALE GROWTH

At this point in the analysis, our sphere map has achieved a configuration generally representative of the shapes within the image, but still retains adverse effects from noise and sub-optimal boundary detection. Although Step 3 results in a reasonably accurate $r(\mathbf{x})$, some spheres still may not reach boundaries, due to pixels incorrectly labeled as outposts. These false outposts will stop spheres in the interior of image regions, leading to the jagged effect on boundaries in the radius images in Figures 16C and 17C. Since the issue at hand is spheres stopping their growth short of the nearest boundary, an added measure to force spheres to grow maximally within their object regions is required. To facilitate this growth variance is used a second time, calculated in the same manner as in Step 2, but used to force growth, rather than limiting it.

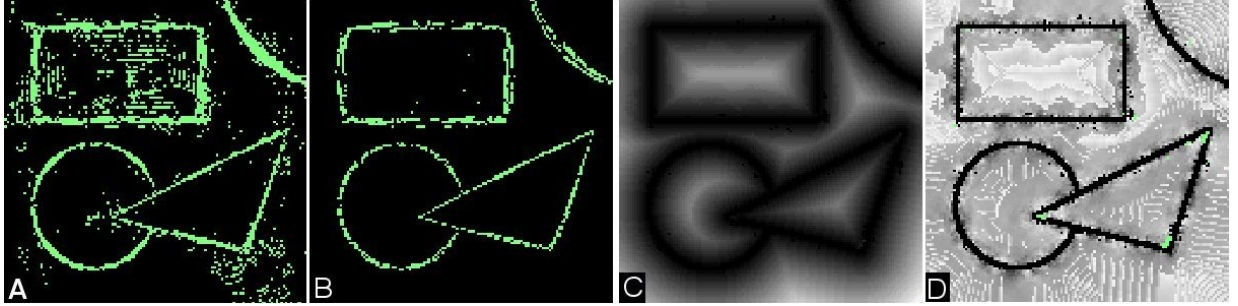


Figure 19: A: Reflector count of all pixels after Step 4 with a binary threshold ($|K^{-1}(\mathbf{x})| \geq 1$) B: Reflector count of all pixels after Step 4 with a binary threshold $|K^{-1}(\mathbf{x})| \geq 4$. C: Radius image $r(\mathbf{x})$ of the image after Step 4. D: Variance after Step 4, with a grayscale range of 0-300.

The global variance threshold α , recalculated as described in Section 4.4, can be used to smooth the boundaries in the radius image by forcing spheres to grow up to the actual boundary using the current, more accurate variance representative of the regions. Due to further optimization of the sphere radii, α represents a more appropriate variance threshold than it did in Step 2, allowing Step 4 to perform more sensitive region-approximating operations. Invoking the variance threshold, the value of $r(\mathbf{x})$ is incremented for all spheres

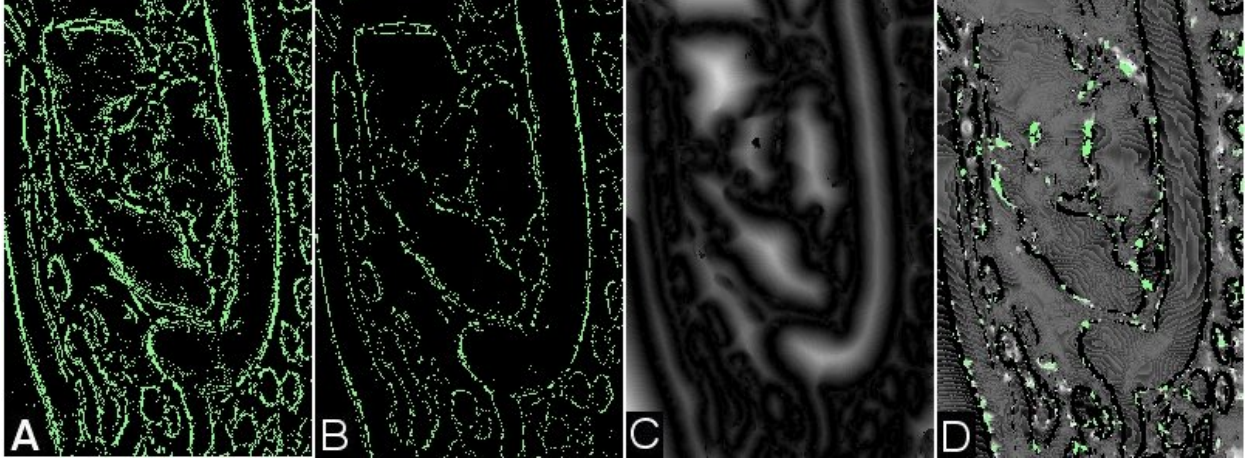


Figure 20: A: Reflector count of all pixels with a binary threshold ($|K^{-1}(\mathbf{x})| \geq 1$) B: Reflector count of all pixels after Step 4 with a binary threshold $|K^{-1}(\mathbf{x})| \geq 4$. C: Radius image $r(\mathbf{x})$ of the image after Step 4. D: Variance after Step 4, with a grayscale range of 0-915.

while $\sigma^2(\mathbf{x}) < \alpha$. This final measure creates a radius image defining spheres which more accurately match the contours of the objects in the image.

The dramatic effects of this final measure can be seen in Figs. 19 and 20 for our synthetic and real images, respectively. It should be noted that in addition to smoothing the $r(\mathbf{x})$ image, this step also sharpens the boundaries (especially apparent in Fig. 19D when compared with Fig. 16D) by causing spheres to grow as far as possible within their object regions.

This last step produces the final optimized set of radius values for each sphere in the image, for use in the segmentation routine to be described in the following section. At this point $r(\mathbf{x})$ is considered optimized.

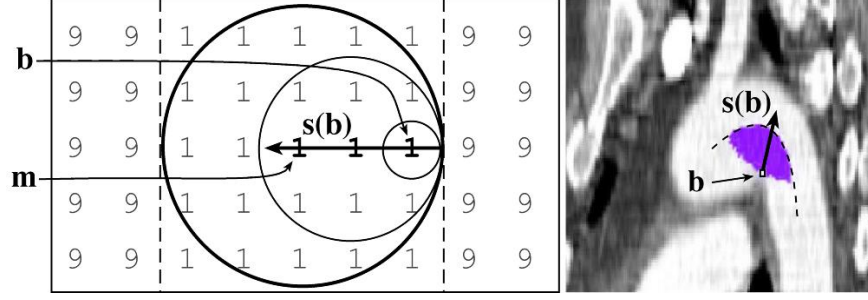


Figure 21: Diagram on the **left** of an object with intensity 1 between two regions of intensity 9, showing set $S^{-1}(\mathbf{b})$ of pixels (bold) whose spheres contain pixel \mathbf{b} . This set produces an $\mathbf{s}(\mathbf{b})$ vector along which the furthest bold pixel \mathbf{m} is the center of a medial sphere (circle in bold) touching both boundaries (dashed lines). Image on the **right** of a 2D slice through a CT scan of the aorta with contrast shows an actual $S^{-1}(\mathbf{b})$ set (purple/grey), the resulting $\mathbf{s}(\mathbf{b})$ vector, and the medial manifold (dashed curve) on which the furthest sphere along $\mathbf{s}(\mathbf{b})$ must lie.

4.5 STEP 5: MEDIAL PIXEL IDENTIFICATION

Given an optimized radius image $r(\mathbf{x})$, the next goal is to extract medial pixels. To facilitate this, a dense set of boundary pixels B is first defined from the sphere map as those pixels with radius 0 or 1,

$$B = \{\mathbf{x} : r(\mathbf{x}) \leq 1\}. \quad (4.9)$$

The sets $S^{-1}(\mathbf{b})$ for all $\mathbf{b} \in B$ can be used to find pixels on the medial manifold, whose spheres are as big as possible while still lying completely within the object. Recall that the $S^{-1}(\mathbf{b})$ set for pixel \mathbf{b} contains all spheres that contain pixel \mathbf{b} . Given a correct sphere map, this set will necessarily contain at least one sphere that touches both the boundary that pixel \mathbf{b} borders as well as an opposing boundary across the object region (and also across the sphere) from pixel \mathbf{b} . Figure 21(left) shows such a medial pixel (labeled “m”) on the medial manifold of an object of intensity 1, between two regions of intensity 9.

To find such medial pixels within $S^{-1}(\mathbf{b})$, we first define an orientation $\mathbf{s}(\mathbf{b})$ orthogonal

to the boundary as the vector sum of the normalized offsets relative to \mathbf{b} for pixels within $S^{-1}(\mathbf{b})$ as

$$\mathbf{s}(\mathbf{b}) = \frac{1}{|S^{-1}(\mathbf{b})|} \sum_{\mathbf{y} \in S^{-1}(\mathbf{b})} \frac{\mathbf{y} - \mathbf{b}}{|\mathbf{y} - \mathbf{b}|}. \quad (4.10)$$

For each boundary pixel $\mathbf{b} \in B$, the pixel $\mathbf{m} \in S^{-1}(\mathbf{b})$ furthest from the boundary along $\mathbf{s}(\mathbf{b})$ is identified as a medial pixel, as depicted in Fig. 21(left). The set of all medial pixels M is thus

$$M = \{\mathbf{m} : \mathbf{m} = \underset{\mathbf{y} \in S^{-1}(\mathbf{b})}{\operatorname{argmax}}((\mathbf{y} - \mathbf{b}) \cdot \mathbf{s}(\mathbf{b})), \mathbf{b} \in B\}. \quad (4.11)$$

Figure 21(right) shows an actual $S^{-1}(\mathbf{b})$ set for a pixel \mathbf{b} on the boundary of the aorta in a CT scan with contrast.

Selecting a single pixel from each $S^{-1}(\mathbf{b})$ set overlooks a large number of additional medial pixels on the outer edge of each set, but no reliable method has been found for extracting only medial pixels from an $S^{-1}(\mathbf{b})$ set. This is in part due to the high variation in shape of $S^{-1}(\mathbf{b})$ sets, which is a product of the local variability of the image objects in question. A standard distance from the boundary for each medial pixel cannot be assumed, as this distance is dependent upon the local thickness of the object and the relative orientation of its boundaries. One can, however, be certain that each $S^{-1}(\mathbf{b})$ set contains a minimum of one medial pixel, as the center of the largest sphere in the direction of the nearest boundary is necessarily a medial pixel. The set M derived taking advantage of this fact is a sparse but reliable set of pixels on the various medial manifolds within the image.

4.6 STEP 6: MEDIAL FLOOD-FILL SEGMENTATION

To segment a particular object, a medial seed pixel $\mathbf{p} \in M$ on that object's medial manifold is manually selected. A flood fill operator is then used to find a connected subset $C \subseteq M$ containing medial pixels that are connected to \mathbf{p} . Pixels belonging in C are found iteratively using a series of sets C_i starting with C_0 , a set containing just the seed pixel \mathbf{p} . At each subsequent step $i + 1$, the set C_{i+1} is created by adding medial pixels within a radius ϕ of

pixels already in set C_i . More precisely, C_i is defined inductively as

$$C_0 = \{\mathbf{p}\} \quad (4.12)$$

$$C_{i+1} = \{\mathbf{x} : \mathbf{x} \in M, S_\phi(\mathbf{x}) \cap C_i \neq \emptyset\}. \quad (4.13)$$

For the results presented, scale ϕ was dynamically set to $\phi(\mathbf{x}) = r(\mathbf{x})/2$, as this causes the algorithm to search half-way from the medial manifold to the boundary for new medial pixels to include, therefore staying within the designated object. When a final step f adds no new pixels, such that

$$C_f = C_{f-1}, \quad (4.14)$$

the flood-fill is complete, as the set of connected medial pixels within the object is the current pixel collection, or

$$C = C_f. \quad (4.15)$$

The union of the set of spheres centered at these medial pixels effectively segments the object by including all of the pixels designated as within the object. These spheres, centered on the medial manifold, extend to all points on the boundary.

5.0 RESULTS

All of the tests shown thus far were run in 2D, for ease in illustration and interpretation. Long term goals for Shells and Spheres, however, involve application to 3D medical image data.

Tests were performed on a 3D contrast-enhanced CT scan containing the aortic arch to showcase performance on real images. Results can be seen in Fig. 22. Figure 22A shows a 2D sagittal slice through the raw 3D data. Figure 22B is the optimized sphere map $r(\mathbf{x})$ of this 2D slice after step 4 showing dark ridges along the boundaries and bright ridges along the medial manifolds. The *mean of means* image $\mu_\mu(\mathbf{x})$, obtained using spheres from the optimized sphere map, is shown in Fig. 22C. A reduction in noise within objects, without significant blurring of boundaries, is evident. Fig. 22D shows the corresponding medial pixels in set M as blue/dark spots and the segmentation from a seed pixel within the aorta as pink/dark shading.

Figure 23 shows the results of the algorithm on the full 3D CT data set from which the example 2D CT images shown throughout this thesis were taken. The surface model shown is simply the union of all spheres in the segmentation set C calculated from a medial seed pixel within the aortic arch. The lighting on the surface model was produced using OpenGL to render the spheres in set C . Figure 24 shows a different 3D CT aortic arch data set, with and without an overlaid segmentation produced by the system presented.

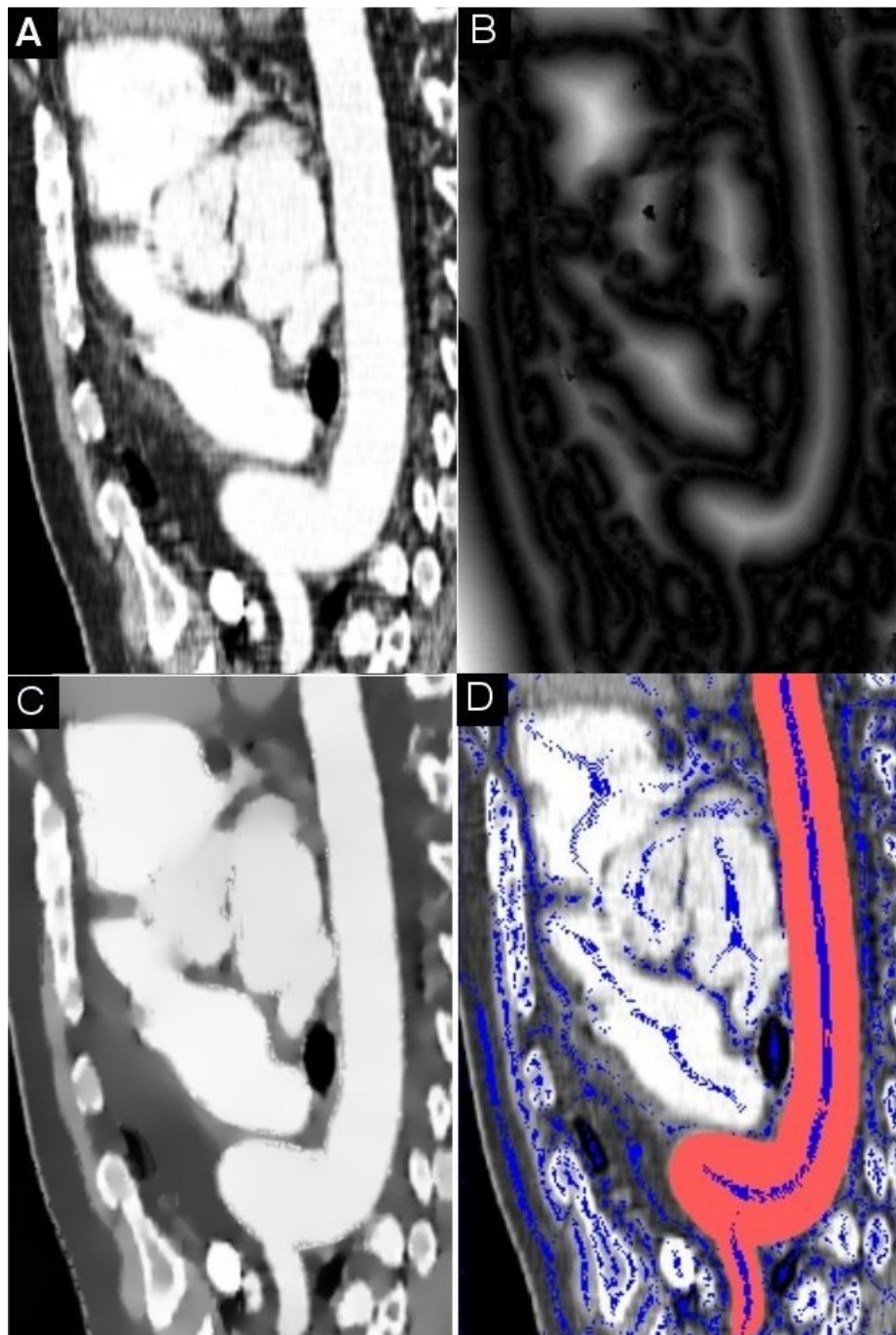


Figure 22: A: 2D CT image of the aortic arch. B: Optimized $r(\mathbf{x})$ image. C: Measured mean of means $\mu_\mu(\mathbf{x})$ image given optimized $r(\mathbf{x})$. D: Segmentation of aortic arch (pink) and medial pixels within the image objects (blue).

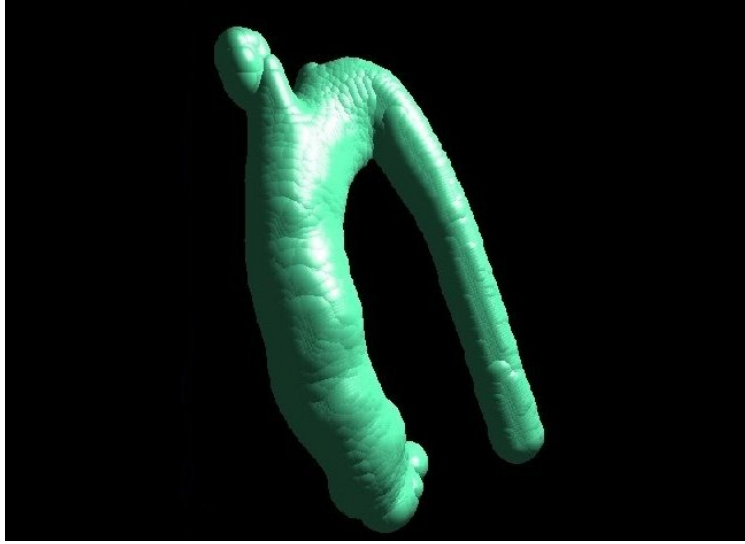


Figure 23: Surface model of the aortic arch produced by applying our algorithm to a 3D contrast-enhanced CT scan, using a single manually placed seed point.

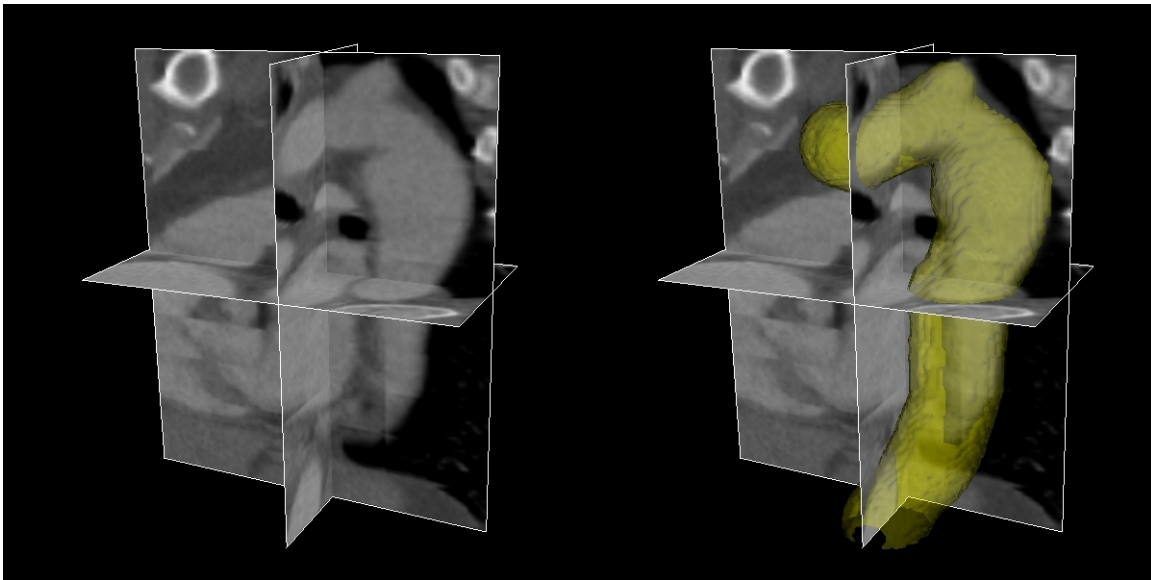


Figure 24: Left: A different 3D contrast-enhanced CT scan of the aortic arch. Right: An overlaid surface model of the aortic arch produced by applying our algorithm, using a single manually placed seed point.

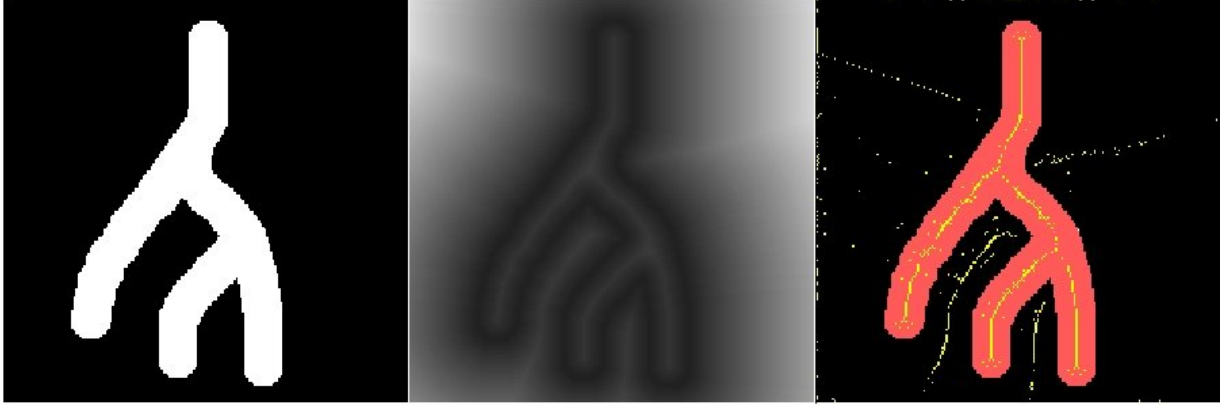


Figure 25: Left: A binary synthetic branching structure image. Middle: The radius image $r(\mathbf{x})$ resulting from optimization using the algorithm presented. Right: A subsequent segmentation of the image, showcasing the medial pixels detected (highlighted in yellow). (DSC = 0.999927)

Branching structures often provide difficulty for medial manifold detection algorithms. A synthetic 2D branching image was used to validate basic performance of the algorithm on detecting and following branching medial structures. The 200x200 test image used and the results obtained are shown in Figure 25. Segmentation performance was quantified by using the Dice Similarity Coefficient (DSC) to compare the segmented region to the known pixels comprising the object in the test image. It can be seen that the branching medial manifold is properly detected, which leads to a successful segmentation with a DSC value of 0.999927 to the correct object. Notice that appropriate medial manifolds are detected outside, as well as inside, the object.

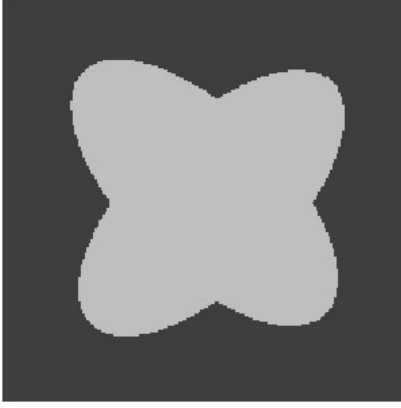


Figure 26: Synthetic binary image used to test segmentation performance in the presence of noise.

To test performance degradation in various levels of noise, the binary synthetic image (black = 0, white = 255) in Figure 26 was used. It should be noted that the object in the image is not perfectly symmetrical. Tests using the optimization algorithm presented were run on this image with increasing amounts of additive uniform noise, added as

$$f(x) = f(x) + N \cdot R - N/2 \quad (5.1)$$

where N is a noise amplitude parameter and R is a random number generated between 0 and 1 using `std::rand()/RAND_MAX`, for all $f(x) \in \Omega$. This produces a uniform random distribution of additive noise over $[-N/2, N/2]$.

Figure 27 shows the images with noise added, the resulting radius images calculated by the algorithm previously described, and resulting the medial manifold and segmentation. Segmentations were all performed with seed points from the center of the object. Figure 28 shows the results of segmentation performance testing in graph form. Performance remains fairly constant (>0.99 DSC) until noise levels above an amplitude of $N = 200$.

The final test, for reference, represents random noise added to each pixel in the range of $[-250 \ 250]$, while the signal of the image itself is only 0-255. While the human visual system is highly skilled at perceiving correlated signals of this sort in high levels of noise, computational systems generally have a great deal of difficulty in such cases. It should also



Figure 27: Synthetic test image with increasing levels of additive noise, split into two columns of test data. The leftmost images in each column are the noisy images, with the amplitude of the additive noise marked in the lower left-hand corner. The middle images in each column are their respective computed radius images. The rightmost images in each column are the segmentations produced along with highlighted medial pixels (yellow), labeled with DSC similarity values for the segmentation.

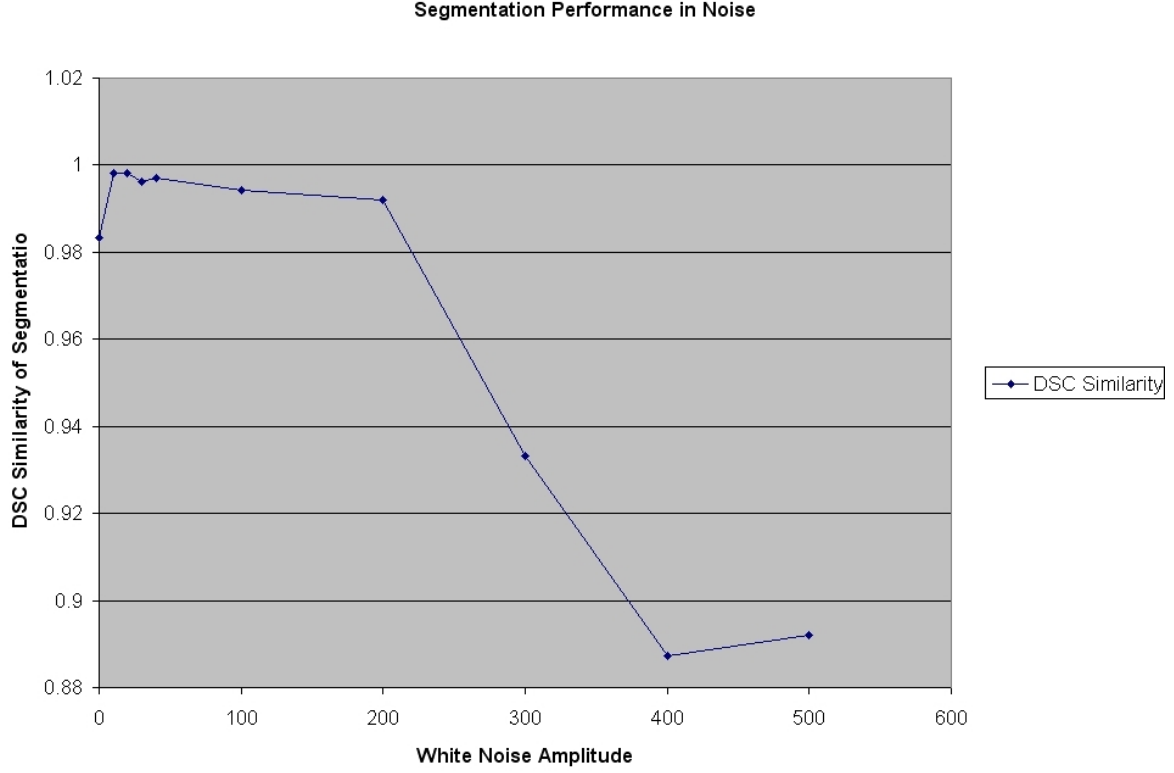


Figure 28: Graph of the amplitude of noise added to the test images versus the segmentation performance, measured by DSC similarity to the known segmentation.

be noted that the level of relative noise in real images intended for application, such as the CT scans show earlier, have lesser amounts of noise than the difficult test images in this study, instead being roughly analogous to the test images between $N = 40$ and $N = 100$.

Run time for the algorithm on 2D images (512x512) was typically 1-2 minutes on a 2.4 GHz Pentium 4-M laptop with 1 gigabyte of RAM. Run time for 3D images (155x90x75) was typically 3-4 hours using a single processor and 2 gigabytes of RAM on a 2 GHz 64-bit quad-Opteron workstation.

6.0 CONCLUSIONS

Shells and Spheres provides an n -dimensional framework for computing Variable Scale Statistics (VSS), using local pixel populations whose size and shape conform to objects in a particular image. The populations are assembled from spheres whose radii vary from pixel to pixel, providing the benefits of rotational invariance without sacrificing the ability to analyze arbitrarily shaped regions. Separate regions span the entire extent of objects on both sides of the boundary, rather than being limited to a small neighborhood near the boundary.

Shells and Spheres differs from multi-scale approaches based on isotropic Gaussian filters in that our domains are not weighted toward zero at their extremities, thus remaining “sharp” at large scale. Our approach is similar to *anisotropic diffusion*[19], in that we seek to avoid blurring boundaries while considering large pixel populations.

The radius image $r(\mathbf{x})$ is analogous to what is often called the *distance map*, which has been well explored for pre-segmented images [1]. The goal of the Shells and Spheres system is to converge on an optimal $r(\mathbf{x})$ on unsegmented grayscale images. A single optimal radius is determined for each pixel such that its sphere touches but does not cross the nearest boundary. In the radius image $r(\mathbf{x})$, troughs represent boundaries, where $r(\mathbf{x})$ is zero. Ridges represent the medial manifold, where $r(\mathbf{x})$ is the medial scale.

The Shells and Spheres framework and VSS operators have been presented, along with one algorithm to optimize $r(\mathbf{x})$ and segment the aortic arch in 2D and 3D from a single seed point. The framework has been shown to be computationally tenable. The progression of the $r(\mathbf{x})$ values, as well as other pertinent statistical values, have been shown on both real and synthetic images through the steps of the given algorithm.

The algorithm presented represents a working, if not optimal, system for demonstrating the concept and potential of the Shells and Spheres framework. While it is believed that

the present algorithm requires too many parameters, the underlying framework offers many possible avenues for further development. The algorithm has been shown to perform well on synthetic images of different types, including branching structures, as well as contrast enhanced aortic arch CT data.

Results of preliminary testing indicate high DSC similarity values for segmentations of noisy synthetic images with known optimal segmentations up to a noise amplitude of > 200 . The capability for accurate segmentation in the presence of noise and even given highly sub-optimal $r(\mathbf{x})$ images indicates a notable level of stability for the application of this approach to segmentation. This advantage is mostly due to the large number of spheres used to define the image and their high degree of overlap within regions, allowing for compensation of lower percentages of incorrectly sized spheres.

Further work is planned to automate the determination of some system parameters. It is hoped that parameters like α , the global variance threshold, can be localized using preliminary region approximation data to improve accuracy. Further work will also be done in developing entirely new algorithms using the statistical framework presented, in hopes of achieving more powerful and computationally efficient future systems.

7.0 ACKNOWLEDGEMENTS

Ken Rockot, Robert Tamburo, John Galeotti, and Dr. George Stetten were all integral parts of the research presented in this thesis, and have my most sincere appreciation and respect.

I would like to thank Dr. Stetten independently for his tremendous guidance, support, and patience throughout my time with VIA Lab.

I would like to express my appreciation to Dr. Sanjeev Schroff and the NIH for their fiscal support through the Cardiovascular Bioengineering Training Program, NIH-T32-HL76124.

I would like to thank the University of Pittsburgh Department of Bioengineering for its support of my education and research throughout the years.

I would also like to thank the University of Pittsburgh Computer Science Department and my co-advisor, Dr. Milos Hauskrecht, for the education and aid I have received through my MS education.

Software was implemented using the Insight Toolkit (ITK), the Visualization Toolkit (VTK), and the Fast Light Toolkit (FLTK).

The CT data used was graciously contributed by Robert Gorman and Michael Sacks.

BIBLIOGRAPHY

- [1] Per Erik Danielsson, “Euclidean distance mapping,” *Computer Graphics and Image Processing*, vol. 14, no. 3, pp. 227–248, Nov. 1980.
- [2] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [3] James Albert Sethian, *Level set methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science*, Number 3 in Cambridge monographs on applied and computational mathematics. Cambridge University Press, Cambridge, U.K., 1996, 218 pages.
- [4] H. Blum and R. N. Nagel, “Shape description using weighted symmetric axis features,” *Pattern Recognition*, 10, pp. 167–180, 1978.
- [5] Tony Lindeberg, *Scale-Space Theory in Computer Vision*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.
- [6] Tony Lindeberg, *Discrete Scale-Space Theory and the Scale-Space Primal Sketch*, Ph.D. thesis, Department of Numerical Analysis and Computer Science, KTH, Stockholm, Sweden, May 1991.
- [7] Jayaram K. Udupa and Supun Samarasekera, “Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation,” *Graphical Models in Image Processing*, vol. 58, no. 3, pp. 246–261, May 1996.
- [8] Jean Serra, “Introduction to mathematical morphology,” *Computer Vision, Graphics, and Image Processing*, vol. 35, no. 3, pp. 283–305, Sept. 1986.
- [9] Anil K. Jain, *Fundamentals of digital image processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.
- [10] Robert M. Haralick and Linda G. Shapiro, *Computer and Robot Vision*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.
- [11] Gunilla Borgefors, “Distance transformations in digital images,” *Computer Vision, Graphics, and Image Processing*, vol. 34, no. 3, pp. 344–371, June 1986.

- [12] O. Cuisenaire and B. Macq, “Fast Euclidean distance transformation by propagation using multiple neighborhoods,” *Computer Vision and Image Understanding*, vol. 76, no. 2, pp. 163–172, Nov. 1999.
- [13] Hinnik Eggers, “Two fast Euclidean distance transformations in Z^2 based on sufficient propagation,” *Computer Vision and Image Understanding*, vol. 69, no. 1, pp. 106–116, Jan. 1998.
- [14] F. Leymarie and Martin D. Levine, “Fast raster scan distance propagation on the discrete rectangular lattice,” *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 55, no. 1, pp. 84–94, Jan. 1992.
- [15] Ingemar Ragnemalm, “Neighborhoods for distance transformations using ordered propagation,” *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 56, no. 3, pp. 399–409, Nov. 1992.
- [16] Punam K. Saha, Jayaram K. Udupa, and Dewey Odhner, “Scale-based fuzzy connected image segmentation: Theory, algorithms, and validation,” *Computer Vision and Image Understanding*, vol. 77, no. 2, pp. 145–174, Feb. 2000.
- [17] Punam K. Saha and Jayaram K. Udupa, “Scale-based diffusive image filtering preserving boundary sharpness and fine structures,” *IEEE Transactions on Medical Imaging*, vol. 20, no. 11, pp. 1140–1155, 2001.
- [18] Jean Serra, *Image Analysis and Mathematical Morphology*, Academic Press, Inc., Orlando, FL, USA, 1983.
- [19] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.